

日本国特許庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出願年月日

Date of Application:

2002年12月27日

出願番号

Application Number:

特願2002-381786

[ST.10/C]:

[JP2002-381786]

出願人

Applicant(s):

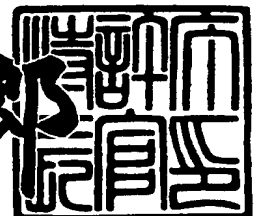
三菱電機株式会社

三菱電機エンジニアリング株式会社

2003年 2月 4日

特許庁長官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2003-3004442

【書類名】 特許願

【整理番号】 542264JP01

【提出日】 平成14年12月27日

【あて先】 特許庁長官 殿

【国際特許分類】 G06F 17/50

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社  
社内

【氏名】 長澤 秀昭

【発明者】

【住所又は居所】 東京都千代田区丸の内二丁目2番3号 三菱電機株式会社  
社内

【氏名】 岡本 泰

【発明者】

【住所又は居所】 東京都千代田区大手町二丁目6番2号 三菱電機エンジニアリング株式会社内

【氏名】 村井 正嗣

【発明者】

【住所又は居所】 東京都千代田区大手町二丁目6番2号 三菱電機エンジニアリング株式会社内

【氏名】 長谷川 直樹

【特許出願人】

【識別番号】 000006013

【氏名又は名称】 三菱電機株式会社

【特許出願人】

【識別番号】 591036457

【氏名又は名称】 三菱電機エンジニアリング株式会社

【代理人】

【識別番号】 100066474

【弁理士】

【氏名又は名称】 田澤 博昭

【選任した代理人】

【識別番号】 100088605

【弁理士】

【氏名又は名称】 加藤 公延

【手数料の表示】

【予納台帳番号】 020640

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 自動回路設計装置およびプログラム

【特許請求の範囲】

【請求項 1】 集積回路内の複数の回路部品に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した表形式のデータ集合を解析する解析部と、

前記解析部の解析結果に基づいて、ハードウェア記述言語による集積回路の記述を生成する記述生成部とを備えた自動回路設計装置。

【請求項 2】 解析部の解析結果に基づいて、記述生成部は、接続条件に応じた各回路部品の接続関係を実現するのに適切なセクタモジュールを選定し、集積回路の記述に前記セクタモジュールに関する情報を付加することを特徴とする請求項 1 記載の自動回路設計装置。

【請求項 3】 解析部の解析結果に基づいて、記述生成部は、セクタモジュールを制御する制御信号を生成するのに適切な制御信号生成モジュールに関する情報を集積回路の記述に付加する請求項 2 記載の自動回路設計装置。

【請求項 4】 集積回路を通常使用する通常使用モードと集積回路を副次的に使用する複数の副次的モードのそれぞれを接続条件とした表形式のデータ集合を解析部が解析し、

前記解析部の解析結果に基づいて、記述生成部は、前記通常使用モードで接続される複数の回路部品の中に介在する第 1 のセクタモジュールを選定し、前記複数の副次的モードでの各回路部品の接続関係を実現するのに適切な第 2 のセクタモジュールを選定し、

前記第 2 のセクタモジュールの出力部が前記第 1 のセクタモジュールの入力部に接続されるように、前記記述生成部が集積回路の記述に前記第 1 のセクタモジュールと前記第 2 のセクタモジュールに関する情報を付加することを特徴とする請求項 2 または請求項 3 記載の自動回路設計装置。

【請求項 5】 集積回路を通常使用する通常使用モードと集積回路を副次的に使用する副次的モードのそれぞれを接続条件とした表形式のデータ集合を解析部が解析し、

前記解析部の解析結果に基づいて、前記副次的モードで接続される複数の回路部品の中に前記副次的モードで信号を有効に伝達するゲートモジュールを記述生成部が選定し、前記記述生成部が集積回路の記述に前記ゲートモジュールに関する情報を付加することを特徴とする請求項 1 から請求項 4 のうちのいずれか 1 項記載の自動回路設計装置。

【請求項 6】 接続条件と接続関係に加えて、各回路部品への給電の停止が可能か否かを記述したデータ集合を解析部が解析し、

前記解析部の解析結果に基づいて、給電の停止が可能な回路部品と他の回路部品との間に配置されて、これらの回路部品の中の信号の伝達を制御するゲートモジュールを記述生成部が選定し、前記記述生成部が集積回路の記述に前記ゲートモジュールに関する情報を付加することを特徴とする請求項 1 から請求項 5 のうちのいずれか 1 項記載の自動回路設計装置。

【請求項 7】 集積回路内の複数の回路部品に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した上位階層の表形式のデータ集合と、各前記回路部品内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の複数の表形式のデータ集合を解析部が解析し、

前記解析部の解析結果に基づいて、記述生成部は、集積回路全体および前記各回路部品に関する総合的な記述を一度に生成することを特徴とする請求項 1 から請求項 6 のうちのいずれか 1 項記載の自動回路設計装置。

【請求項 8】 コンピュータを、

集積回路の複数の回路部品に関する接続条件と前記接続条件に応じた各回路部品の接続関係とを記述した表形式のデータ集合を解析する解析部と、

前記解析部の解析結果に基づいて、ハードウェア記述言語による集積回路の記述を生成する記述生成部として機能させるためのプログラム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

この発明は、自動回路設計装置およびプログラムに関するものである。

## 【0002】

## 【従来の技術】

近年、大規模集積回路（LSI）に対する要求仕様は複雑化し回路規模も増大している。そのため、ゲートレベルネットリストの設計よりも、上位の言語レベルであるハードウェア記述言語（HDL）による設計が広く行われるようになってきている。

## 【0003】

集積回路（LSI）の技術の進展に伴い、システムLSIを構成する機能ブロックが、LSIを設計するための共有資産であるIP（intellectual property）として流通および再利用されている。複数の既存のIPと他の既存の回路部品を組み合わせることにより、LSIの全体（トップ階層回路）またはモジュールを設計することが可能であり、設計作業が容易になっている。一般に流通しているIPは、HDLのうちレジスタトランスファレベル（RTL）で記述されたものが多い。

## 【0004】

## 【発明が解決しようとする課題】

しかし、このような設計でも、IPまたはその他の回路部品を接続するには、集積回路が作動する条件（モード）に応じて各回路部品へ供給される信号の相違を想定しながら、セクタモジュールやセクタモジュールを制御する制御信号生成モジュールなどの追加すべき回路部品を適切に配置するなどの複雑な思考が設計者に要求されている。このような諸事情を考慮しながら、設計者は集積回路の記述（ソース）を作成している。通常、IPまたはその他の回路部品を利用し、必要な設計仕様に基づいて実際の集積回路の記述を作成するには、HDL、特にレジスタトランスファレベルに関する知識を必要とする。従って、限られた人にしかLSIの設計ができなかった。

## 【0005】

現在では各IPの接続ポートの数は増加し、数百を越える接続が必要となることも珍しくない。人手でHDLによる回路記述を書き直していると、かなりの手間と時間をとられ、さらに記述ミスによる接続不具合が発生するおそれも増加す

る。

#### 【0006】

この発明は、上記のような課題を解決するためになされたもので、複数の回路部品を接続した集積回路の自動設計を容易にし、HDLによる回路記述の作成にかかる手間と時間を短縮する自動回路設計装置およびプログラムを得ることを目的とする。

#### 【0007】

##### 【課題を解決するための手段】

この発明に係る自動回路設計装置は、集積回路の複数の回路部品に関する接続条件と前記接続条件に応じた各回路部品の接続関係とを記述した表形式のデータ集合を解析する解析部と、前記解析部の解析結果に基づいて、ハードウェア記述言語による集積回路の記述を生成する記述生成部とを備えたものである。

#### 【0008】

##### 【発明の実施の形態】

以下、この発明の実施の一形態を説明する。

##### 実施の形態1.

図1は、この発明の実施の形態1に係る自動回路設計装置を示すブロック図である。この自動回路設計装置は、ソフトウェアに基づいて動作するコンピュータであるプロセッサ1、記憶部2、表示部3および入力部4を有する。

#### 【0009】

記憶部2は、例えばハードディスクであって、プロセッサ1で使用されるファイル（データ集合）およびプロセッサ1で作成されたファイルを格納する。図示しないが、プロセッサ1はファイルを記憶部2とは別の記憶媒体に格納してもよいし、通信装置により、ファイルを他の装置に送出してもよい。また、プロセッサ1は、これらのファイルに基づいて、表示部3を制御して画像を表示させ、設計者に情報を提示する。設計者は入力部4を用いてプロセッサ1に指示を与えることができる。

#### 【0010】

プロセッサ1は、機能すなわち実行するプログラムモジュールに応じて、便宜

的にスプレッドシート生成部 1 0、解析部 1 1 および記述生成部 1 2 に分類される。次に、プログラムに従ってプロセッサ 1 が実行する動作について図 2 のフローチャートを参照しながら説明する。

#### 【0 0 1 1】

まず、ステップ S T 1 では、スプレッドシート生成部 1 0 が一覧表形式（スプレッドシート形式）の画面を表示部 3 に表示させ、これによって設計者による一覧表への情報記入すなわち一覧表の作成を促す。ここで表示されて設計者に記入された一覧表の例を図 3 に示す。この一覧表の各行は、L S I または L S I の部品となるモジュールの各ピン（パッドまたは入出力セル）の接続情報を表す。第 1 列にはそれらのピンの名前が書かれ、他の列には各ピンに接続される要素が列挙されている。この一覧表は、接続が成立する条件（接続条件 1 ～ 3）に従って列に区切られていて、各ピンがいかなる接続条件でどの要素に接続されるかを表す。ここでいう接続条件とは、例えば L S I の動作のモードが含まれる。

#### 【0 0 1 2】

図 3 の例は、ピン P A、P B、P C、P D が、いかなる接続条件で I P（回路部品）A、B、C のピン I P A 1 ～ I P A 4、I P B 1、I P B 2、I P C 1 ～ I P C 3 のいずれに接続されるのかを表す。具体的には、ピン P A は接続条件 1 では I P A の第 1 のピン I P A 1 に接続され、接続条件 2 では I P B の第 1 のピン I P B 2 に接続され、接続条件 3 では I P C の第 1 のピン I P C 1 に接続されることを図 3 の一覧表は示している。同様に、ピン B は、どの接続条件でも I P A の第 2 のピン I P A 2 に接続されることが示されている。すなわち、この一覧表は、L S I の複数の回路部品に関する接続条件（接続条件 1 ～ 接続条件 3）と、これらの接続条件に応じた各回路部品の接続関係とを記述する。

#### 【0 0 1 3】

スプレッドシート生成部 1 0 は、一覧表のフォーマット（各欄が空いているもの）を表示部 3 に表示させる。設計者は、必要な設計仕様を考慮しつつ入力部 4（例えばマウスのようなポインティングデバイスとキーボード）を用いて、グラフィック表示された一覧表の各欄に情報を記入することができる。このようにして一覧表を作成すると、設計者は入力部 4 を操作して一覧表に対応するスプレッ



ドシートファイルを記憶部 2 に格納することが可能である。

【0014】

理解を容易にするため、図 3 の一覧表に基づいて最終的に設計される目的の回路を図 4 に示す。図 4 から明らかなように、ピン PA, PB は IPA, B, C に信号を入力する要素であり、ピン PC, PD は IPA, B, C から出力される信号を受ける要素である。目的の回路には、接続条件に従って IPA, B, C とピン PA, PB の接続関係を変更するためにセクタモジュール SL1, SL2 およびエンコーダモジュール（制御信号生成モジュール）20 が設けられている。かかるセクタモジュール SL1, SL2 およびエンコーダモジュール 20 は、後述するようにプログラムに従ったプロセッサ 1 の処理により挿入されたものである。

【0015】

図 2 に戻り、一覧表の作成後、解析部 11 が一覧表を読み込んで、その内部のデータを解析する（ステップ ST2）。ここで解析部 11 は、記憶部 2 からスプレッドシートファイルを読み込んでもよいし、記憶部 2 に未保存の一覧表を読み込んでもよい。

【0016】

この後、解析部 11 の解析結果に基づいて、記述生成部 12 が必要なモジュールおよび信号の記述を HDL によって生成し（ステップ ST3～ST7）、その結果得られたモジュールおよび信号の記述を組み合わせた回路記述を出力する（ステップ ST8）。得られた回路記述は表示部 3 で確認することができ、設計者は入力部 4 を操作して HDL による回路記述ファイルを記憶部 2 に格納することが可能である。

【0017】

図 5 は、図 3 の一覧表に基づいて作成された回路記述の一部を抜粋して示すリストである。この実施の形態に関する図 5 およびその他の図面に示される回路記述は、「Verilog-HDL」の RTL に準拠する。一行目は、設計された回路が複数の IP を包含する LSI の全体（トップ階層回路すなわちトップ階層モジュール）であって、ピン PA, PB, PC, PD を有することを示す。二行

目および三行目は、ピン P A、P B が信号を入力する要素であり、ピン P C、P D は出力される信号を受ける要素であることを示す。四行目および五行目は、接続条件 1 が満たされた（条件 1' b1 に合致した）場合には、ピン P A が I P A の第 1 のピン I P A 1 に接続されることを示す。なお、条件 1' b1 が接続条件 1 に対応することは、設計者が指定してある。

## 【 0 0 1 8 】

図 2 に示すように、一覧表の解析に基づいて H D L による必要なモジュールの定義文を生成する工程には、接続条件解析（ステップ S T 3）、セレクトモジュールの挿入（ステップ S T 4）、分岐モジュールの挿入（ステップ S T 5）、特殊モジュールの挿入（ステップ S T 6）、および各 I P の接続設定（ステップ S T 7）がある。ステップ S T 4 ～ S T 7 の工程は必ずしも図 2 のフローチャートに示された順序に従って実行する必要はなく、他の順序で実行してもよいし、複数のステップを並列に実行してもよい。また、ステップ S T 4 ～ S T 7 の工程のいずれかを任意に省略するようにプログラミングしてもよい。

## 【 0 0 1 9 】

次に、L S I の記述を生成する工程の基本的考え方を説明する。図 6 は、設計者が指定した複数の回路部品とこれらの回路部品を接続するための配線を例示する。図示のように、回路部品相互を接続する配線は、一つ以上のモジュールとして捉えることができる。一覧表に記述された接続条件と接続関係に応じて、記述生成部 1 2 は適切なモジュールを選択する。これらのモジュールは、ステップ S T 8 で最終的な回路記述が出力されるまで、プログラムが使用する固有のオブジェクトとしてプロセッサ 1 に保持される。プログラムのステップ S T 3 ～ S T 7 では、図 6 に例示する各ブロックに適当なセル（回路の最小要素）を配置する。このようなセルの配置は、オブジェクトのメソッドの変更に対応する。

## 【 0 0 2 0 】

以下、プログラムの各ステップについて、さらに詳細に説明する。

ステップ S T 2（図 2）の一覧表の読み込みでは、解析部 1 1 は、一覧表の各行の接続情報を抽出する。抽出した接続情報は、解析部 1 1 によりプログラムが使用する固有のオブジェクトとして保管される。ここでは二階層のオブジェクト



が使用される。一階層は各ピンのピン名、各接続条件での接続先のデータを保持するピンオブジェクトであり、もう一階層はピンオブジェクトを複数個保持しており一つの一覧表全体を表すモジュールオブジェクト（またはIPオブジェクト）である。以下の各ステップでは、オブジェクトで使用されたメソッドとしてのモジュールまたは信号の記述を追加または変更する。

#### 【0021】

ステップST3（図2）の接続条件解析では、記述生成部12は、一覧表の各列の接続条件を解析する。具体的には、記述生成部12は、各接続条件でアクティブとなる信号（この信号をモード識別信号と呼ぶ）を確保し、各接続条件とモード識別番号を関連づける条件定義文を作成する。図3のように接続条件（モード）が三通りあるような一覧表については、三つのモード識別信号が定義され、各接続条件に応じて、いずれかのモード識別信号がアクティブとなり、他のモード識別信号が非アクティブとなる。

#### 【0022】

図3および図5に示された接続条件は、実際には、ある信号線がローレベル（L）かハイレベル（H）かを示す条件論理式で表されたり、あるいはいくつかの信号線の各レベルを示す条件論理式で表されたりする。そのような形式で図3の一覧表に記入された条件論理式を解析部11はそのまま保持し、接続条件解析で記述生成部12はこれらの条件論理式とモード識別信号を関連づける条件定義文を作成する。複数のモード識別信号が同時にアクティブになることは避けることが好ましい。従って条件定義文で使用される条件分岐式が互いに排他的である必要がある。最終的には、記述生成部12は、図7に示す条件分岐式を含んだ条件定義文を意味する記述を生成する。

#### 【0023】

ステップST4（図2）のセレクトモジュールの挿入では、記述生成部12は、接続条件に応じた各回路部品の接続関係を実現するのに適切なセレクトモジュール（例えば図4のセレクトモジュールSL1、SL2）を選定し、セレクトモジュールに関する定義文を保持する。また、記述生成部12は、セレクトモジュールを制御する制御信号を生成するのに適切な制御信号生成モジュールに関する

情報をLSIの記述に付加する。

【0024】

具体的には、プログラムに従って記述生成部12は、各回路部品（例えばIPA）の入力ピンに対する接続先を調べるとともに接続先をカウントする。例えば、図3の一覧表では、IPA、B、Cと別の回路部品にとっての入力ピンPCの接続先は、ピンIPA3、IPB2、IPC2の三つであり、ピンPDの接続先はピンIPA4、IPC3の二つである（同じ接続先はカウントしない）。このようにして調べた接続先をプログラムで利用する配列として記述生成部12は保持する。以後、この接続先を表した配列を接続認識配列と呼ぶ。

【0025】

記述生成部12は、接続認識配列と接続先の個数に基づいて適切なセクタモジュールを適切なピンに接続することを記述した接続定義文を生成して、保持する。接続条件の数にかかわらず、各セクタモジュールは、接続先の個数と同数の入力部と、一つの出力部を有する。例えば、ピンPCについては、三つの入力部と一つの出力部を有するセクタモジュールSL1が選択される。

【0026】

仮に入力ピンへの接続先が一つの場合、記述生成部12はセクタモジュールを挿入せずに、単純な接続定義文あるいはバッファのみを記述した文を生成して保持する。例えば、IPAの入力ピンIPA1の接続先は一つのピンPAだけであるので、ピンIPA1とピンPAを接続する旨だけを規定した定義文か、ピンIPA1とピンPAの間にバッファが介在する旨を規定した文を生成して保持する。

【0027】

さらに、記述生成部12は、制御信号生成モジュール（エンコーダモジュール）を定義する。制御信号生成モジュールは、モード識別番号に従ってこれらのセクタモジュールを制御する制御信号を各セクタモジュールに供給する。制御信号生成モジュールは、前記の接続条件と接続認識配列に基づいて定義され、記述生成部12は、制御信号生成モジュールを適切にセクタモジュールに接続することを記述した接続定義文を生成して保持する。

## 【 0 0 2 8 】

この制御信号生成モジュールによって、セレクトアモジュールは接続条件（例えば L S I の動作モード）に合ったパスを選択することができる。また、複数の接続条件について同じ接続が存在しても、余計なセレクトアモジュールを作成せずに、必要最低個数のセレクトアモジュールのみで接続条件に合ったパスの選択が可能になる。

## 【 0 0 2 9 】

実際の L S I では、L S I を通常使用する通常使用モードと L S I を副次的に使用する副次的モード（例えば L S I を試験するテストモード）とでは、一般的に性能に対する要求が異なる。重要性が高い通常使用モードでは、要求される処理速度が高く、電力消費も最小限にすべきである。一方、テストモードは、各回路部品が機能的に動作していることを確認することのみを目的とすることが多く、処理速度、容量、抵抗に関する許容度が、通常使用モードより高い。そこで、複数の副次的モードがある L S I の設計におけるセレクトアモジュールの挿入に関しては、以下のように工夫すると好ましい。

## 【 0 0 3 0 】

まず、図 8 に例示する一覧表をステップ S T 1 で設計者が完成させる。この一覧表は、L S I を通常使用する通常使用モードと L S I を副次的に使用する複数の副次的モードのそれぞれを接続条件として記述し、各モードに応じた回路部品の接続関係を記述する。ステップ S T 3 の接続条件解析では、記述生成部 1 2 は、一覧表の各列のモードとモード識別番号を関連づける条件定義文を作成する。

## 【 0 0 3 1 】

そして、ステップ S T 4 のセレクトアモジュールの挿入では、記述生成部 1 2 は、通常使用モードで接続される複数の回路部品（例えば I P ）の間に介在する第 1 のセレクトアモジュールを選定する。また記述生成部 1 2 は、複数のテストモードでの各回路部品の接続関係を実現するのに適切な第 2 のセレクトアモジュールを選定し、第 2 のセレクトアモジュールの出力部が第 1 のセレクトアモジュールの入力部に接続されるように、第 1 のセレクトアモジュールと第 2 のセレクトアモジュールに関する記述を生成する。また、記述生成部 1 2 は、制御信号生成モジュールを

これらのセクタモジュールに接続することを記述した接続定義文を生成する。

【0032】

図9は図8に例示する一覧表に基づいて最終的に設計される目的の回路を示す。図示の第1のセクタモジュールSL3の入力部は、通常パスNPとテストパスTPに接続されており、あるIPの入力部にセクタモジュールSL3の出力部は接続されている。複数のテストモードが利用されることから、複数のテストパスが必要になる。この設計例では、テストパスTPに出力部が接続された第2のセクタモジュールSL4を設けることにより、テストパスTPを複数のテストパスTP1～TP3に分岐させている。

【0033】

セクタモジュールSL3、SL4を制御するためにエンコーダモジュール（制御信号生成モジュール）21が設けられている。エンコーダモジュール21は単一の信号線BAを介して第1のセクタモジュールSL3に接続されており、通常パスNP上の信号とテストパスTP上の信号のいずれかを選択する制御信号をセクタモジュールSL3に供給する。また、エンコーダモジュール21は複数の並列信号線BBを介して第2のセクタモジュールSL4に接続されており、セクタモジュールSL3にテストパスTP上の信号を選択させている時に、テストパスTP1～TP3のいずれかを選択する制御信号をセクタモジュールSL4に供給する。

【0034】

このように通常パスNPとテストパスTPのために、二つの入力部と一つの出力部を有する第1のセクタモジュールSL3を設け、テストパスTP1～TP3のために、三つの入力部と一つの出力部を有する第2のセクタモジュールSL4を設けている。従って、一つの通常使用モードと複数のテストモードで動作するLSIにおいて一つの通常パスNPと複数のテストパスTPが設けられても、第1のセクタモジュールSL3は、通常パスNPとテストパス群の二つの選択肢しか持たないので、その内部構成（例えば内部ゲートの個数や配線パターン）はセクタのうち最も簡単である。このため通常使用モードでは、高い処理速度と、低い電力消費を達成することが可能である。

## 【 0 0 3 5 】

図 1 0 は図 9 に記載の回路に関する各セクタモジュールの記述を平易化して示すリストであり、図 1 1 は図 9 に記載の回路に関するエンコーダモジュールの記述を平易化して示すリストである。

## 【 0 0 3 6 】

図 1 0 の 1 行目～ 5 行目は、信号線 B A 上の信号が 1 であると、セクタモジュール S L 3 は I P の入力部へ通常バス N P 上の信号を供給し、それ以外の場合には、セクタモジュール S L 3 は I P の入力部へセクタモジュール S L 4 の出力信号を供給することを示す。また、 6 行目および 7 行目は、信号線 B B 上の信号が 1 であると、セクタモジュール S L 4 はテストバス T P 1 上の信号を供給することを示す。 8 行目および 9 行目は、他の場合に、信号線 B B 上の信号が 2 であると、セクタモジュール S L 4 はテストバス T P 2 上の信号を供給することを示す。 1 0 行目および 1 1 行目は、他の場合にはセクタモジュール S L 4 はテストバス T P 3 上の信号を供給することを示す。

## 【 0 0 3 7 】

図 1 1 の 1 行目～ 3 行目は、モード識別信号 1 がオンであると、エンコーダモジュール 2 1 はセクタモジュール S L 3 に通常バス N P を選択する制御信号を信号線 B A に流し、セクタモジュール S L 4 に本来不要であるがテストバス T P 1 を選択する制御信号を信号線 B B に流すことを示す。 4 行目～ 6 行目は、他の場合に、モード識別信号 2 がオンであると、エンコーダモジュール 2 1 はセクタモジュール S L 3 にテストバス T P を選択する制御信号を信号線 B A に流し、セクタモジュール S L 4 にテストバス T P 1 を選択する制御信号を信号線 B B に流すことを示す。 7 行目～ 9 行目は、他の場合に、モード識別信号 3 がオンであると、エンコーダモジュール 2 1 はセクタモジュール S L 3 にテストバス T P を選択する制御信号を信号線 B A に流し、セクタモジュール S L 4 にテストバス T P 2 を選択する制御信号を信号線 B B に流すことを示す。 1 0 行目～ 1 2 行目は、他の場合には、エンコーダモジュール 2 1 はセクタモジュール S L 3 にテストバス T P を選択する制御信号を信号線 B A に流し、セクタモジュール S L 4 にテストバス T P 3 を選択する制御信号を信号線 B B に流すことを示す。

## 【0038】

ステップST5（図2）の分岐モジュールの挿入では、記述生成部12は、解析部11の解析結果に基づいて、副次的モード（例えばテストモード）で接続される複数の回路部品（例えばIP）の間に副次的モードで信号を有効に伝達するゲートモジュールを選定し、記述生成部12がゲートモジュールに関する記述を生成して保持する。設計者は図8に示すような通常使用モードと副次的モードを記述した一覧表を完成しておく。ステップST3の接続条件解析では、記述生成部12は、一覧表の各列のモードとモード識別番号を関連づける条件定義文を作成する。

## 【0039】

図12は、分岐モジュールを具体的に示す回路図である。設計者が完成した一覧表では、一つのIP30の出力ピンが、通常使用モードでは回路部品モジュール31の入力ピンに接続され、テストモードでは回路部品モジュール32の入力ピンに接続されると一覧表で指定されている。ただし、回路部品モジュール31、32は実質的に無内容な単なる接続線である場合もある。

## 【0040】

IP30と、通常使用モードで使用される回路部品モジュール31は、単に通常パスNPで接続すればよい。しかし、通常使用モードでは、IP30から回路部品モジュール32に信号が伝搬しないようにして消費電力の増大を防止する必要がある場合が多い。そこで、最終的に得られる回路が図12に示す構造を持つように、記述生成部12が接続認識配列に基づいてゲートモジュールに関する記述を生成する。この分岐モジュールの挿入のためにも、プログラムに従って記述生成部12は、接続認識配列を生成して保持するが、分岐モジュールの挿入で作成される接続認識配列は、各回路部品（例えばIP）の入力ピンではなく出力ピンに対する接続先を調査した結果である。

## 【0041】

図12に示すように、最終的に得られる回路にはANDゲートモジュール34が設けられており、ANDゲートモジュール34の出力部はテストモードで使用



される回路部品モジュール 3 2 にテストパス T P を介して接続されている。通常パス N P の途中から分岐線が分岐しており、これが A N D ゲートモジュール 3 4 の一方の入力部に接続されている。このように分岐線と A N D ゲートモジュール 3 4 とテストパス T P を持つ分岐モジュール 3 5 が設けられている。

## 【 0 0 4 2 】

A N D ゲートモジュール 3 4 の他方の入力部は、エンコーダモジュール（ゲート制御モジュール） 3 6 に接続されている。エンコーダモジュール 3 6 は、L S I の通常使用モードでは、ローレベルの電位を A N D ゲートモジュール 3 4 に与える。従って、L S I の通常使用モードでは、テストパス T P 上の電位はローレベルに固定され、電力消費が抑えられる。他方、テストモードでは、ハイレベルの電位をエンコーダモジュール 3 6 は A N D ゲートモジュール 3 4 に与え、I P 3 0 から回路部品モジュール 3 2 への実質的な接続を可能にする。エンコーダモジュール 3 6 は、接続条件解析（図 2 のステップ S T 3）で得られた前記の接続条件に基づいて定義される。

## 【 0 0 4 3 】

実際の記述では、A N D ゲートモジュールを指定するには、記述生成部 1 2 は、固定モジュールのインスタンス呼出文を記述する。例えば以下のようにである。

TBKT01(.A(I P の出力ピンの出力信号名の指定),.B(通常モードでローレベルになる信号名の指定),.N(通常パスノードの指定),.T(テストパスノードの指定))  
;

ここでTBKT01とはゲートモジュールの呼び名の例であり、ゲートモジュールの呼び名はこの表記に限らず設計者が任意に決定できる。

## 【 0 0 4 4 】

仮に出力ピンへの接続先が一つのみでありその接続が通常使用モードだけで行われる場合、記述生成部 1 2 は分岐モジュールに A N D ゲートモジュールを配置するのではなくバッファのみを配置した定義文を生成するか、単純な接続定義文を生成する。また、出力ピンへの接続先が一つのみでありその接続がテストモードだけで行われる場合、その出力ピンには通常パスが接続されずテストパスのみ

が接続されるが、そのテストパスの途中にはANDゲートモジュールが配置され、ANDゲートモジュールの一方の入力端がエンコーダモジュールに接続される。この場合も、エンコーダモジュールは、通常使用モードでは、ローレベルの電位をANDゲートモジュールに与え、テストモードでは、ハイレベルの電位をANDゲートモジュールに与える。これにより、テストモードでは、出力ピンからANDゲートモジュールの出力部への実質的な接続が可能になる一方、通常使用モードではANDゲートモジュールの出力部の電位はローレベルに固定され、電力消費が抑えられる。

## 【 0 0 4 5 】

一つの出力ピンに接続されるテストパスが複数ある場合は、このANDゲートモジュールの出力部を他のANDゲートモジュールの一方の入力部に接続し、各ANDゲートモジュールをエンコーダモジュールで制御すればよい。

## 【 0 0 4 6 】

ステップST6（図2）の特殊モジュールの挿入では、記述生成部12は、解析部11の解析結果に基づいて、給電の停止が可能な回路部品（例えばIP）と他の回路部品（例えばIP）との間に配置されて、これらの回路部品の間の信号の伝達を制御するゲートモジュールを選定し、記述生成部12がゲートモジュールに関する情報を生成して保持する。この特殊モジュールの挿入は、前述した接続条件と接続関係に加えて、各回路部品への給電の停止が可能か否かを記述したファイルを解析部11が解析した場合に実行される。従って、各回路部品への給電の停止が可能か否かを記述した一覧表をステップST1で設計者が完成させていることが、特殊モジュールの挿入の前提条件である。ステップST2の一覧表の読み込みでは、解析部11は、各回路部品への給電の停止が可能か否かを示す情報を保持する。

## 【 0 0 4 7 】

図13は、特殊モジュールの挿入で付加されたゲートモジュールを具体的に示す回路図である。図に示すように、同一チップにおいて境界BDを間に挟んでブロックBL1、BL2が存在する。ブロックBL1はLSIの動作中に給電の停止が可能なブロックであり、ブロックBL2はLSIの動作中は常に給電されて

いるブロックである。ブロック B L 1 の通常使用モードでは、ブロック B L 1 中の I P 4 0 から通常パス N P 1 を介してブロック B L 2 中の I P 4 1 に信号が供給される必要がある。

## 【 0 0 4 8 】

ブロック B L 1 には、ブロック B L 1 のテストモードでの消費電力増大防止のために分岐モジュール 4 2 が設けられている。すなわち、通常パス N P 1 から分岐したの途中から分岐線が分岐しており、これが A N D ゲートモジュール 4 3 の一方の入力部に接続されている。A N D ゲートモジュール 4 3 の他方の入力部にハイレベルの電位が与えられた場合に限って、A N D ゲートモジュール 4 3 はその出力部に接続されたテストパス T P 4 に I P 4 0 からのテスト信号を供給することができるようになっている。

## 【 0 0 4 9 】

他方、ブロック B L 2 には、ブロック B L 2 の通常使用モードでの動作とテストモードでの信号の流れを区別するためのセクタモジュール S L 5 が配置されている。セクタモジュール S L 5 は二つの入力部を有しており、これらの入力部はそれぞれ通常パス N P 2 とテストパス T P 5 に接続されている。

## 【 0 0 5 0 】

図示のように、L S I の動作中に給電の停止が可能なブロック B L 1 から L S I の動作中に常に給電されている（給電の停止が不可能な）ブロック B L 2 へ信号が供給されるチップにおいては、ブロック B L 1 への給電を停止しても、実際には暫くの間、I P 4 0 から I P 4 1 への信号の電位がローレベルにならず、このため中間電位のリーク電流が I P 4 1 に悪影響を及ぼす可能性がある。そこで、図示の実施の形態では、I P 4 0 と I P 4 1 との間に A N D ゲートモジュール（特殊モジュール）4 4 が配置されている。

## 【 0 0 5 1 】

A N D ゲートモジュール 4 4 の一方の入力部は、I P 4 0 の出力信号が流れる通常パス N P 1 に接続され、他方の入力部は制御パス C P に接続されている。また、A N D ゲートモジュール 4 4 の出力部は、I P 4 1 に入力される信号が流れる通常パス N P 2 に接続されている。このような構成の下、制御パス C P の電位

がハイレベルの場合に限って、ANDゲートモジュール44の出力部に接続された通常パスNP2での電位がハイレベルになりうる。従って、ブロックBL1への給電停止と同時に制御パスCPでの電位をローレベルとするように制御すれば、ブロックBL1への給電停止後に、通常パスNP2の電位は常にローレベルに固定され、IP41に不適切な信号が供給される事態にはならない。

## 【0052】

LSIの構成および動作モードによっては、逆に、ブロックBL1への給電停止の後でも、ブロックBL2のIP41への入力信号の電位が常にハイレベルである方が望ましい場合も考えられる。このような場合には、ANDゲートモジュール44に代えてORゲートモジュール（図示せず）を配置してもよい。この構成では、ブロックBL1への給電停止と同時に制御パスCPでの電位をハイレベルとするように制御すれば、ブロックBL1への給電停止後に、通常パスNP2の電位は常にハイレベルに固定され、やはりIP41に不適切な信号が供給される事態にはならない。

## 【0053】

このように、LSIの動作中に給電の停止が可能な回路部品出力部と他の回路部品との間に配置されて、給電の停止が可能な回路部品からの信号の伝達を制御する（すなわちその回路部品からの出力電位を固定する）ゲートモジュールを特殊モジュール（首締めブロック）と呼び、このブロックを有する接続を特殊接続と呼ぶ。

## 【0054】

また、LSIの動作中に常に給電されている（給電の停止が不可能な）ブロックから出力されたハイレベルの信号が、給電が停止されているブロックのpチャネルMOSトランジスタのドレインあるいはソースに供給される場合、接合方向が順方向となるために、ソースとドレイン間にリーク電流が発生する可能性がある。そこで、常に給電されている（給電の停止が不可能な）ブロックと、給電の停止が可能なブロックのpチャネルMOSトランジスタのドレインあるいはソースの間に、ゲートモジュール（例えばバッファ）を介在させて、ゲート電圧に従ってドレイン電流が正確に制御されるようにしてもよい。このような目的で配置

されるゲートモジュールも特殊モジュールと呼ぶ。

#### 【 0 0 5 5 】

図 1 3 に示すように、L S I の動作中に給電の停止が可能なブロック B L 1 に分岐モジュール 4 2 が存在し、L S I の動作中に常に給電されている（給電の停止が不可能な）ブロック B L 2 にセクタモジュール S L 5 が存在する場合には、このような特殊接続は、分岐モジュール 4 2 の分岐線とセクタモジュール S L 5 の中間に位置するように配置する。

#### 【 0 0 5 6 】

上述のように、各回路部品（例えば I P）への給電の停止が可能か否かを記述した一覧表をステップ S T 1 で設計者が完成させている。ステップ S T 2 の一覧表の読み込みでは、解析部 1 1 は、一つの一覧表全体を表す I P オブジェクトに、各回路部品への給電の停止が可能か否かを示す情報を保持しておく。この情報に基づいて、常に給電されている（給電の停止が不可能な）ブロックと、給電の停止が可能なブロックの境界 B D を横切る接続線には、特殊モジュールを挿入することになる。またその場合、給電が停止されるとローレベルまたはハイレベルになる制御バス C P 上の信号も I P オブジェクトに記憶しておき、ステップ S T 6 の特殊接続の挿入では、この信号と各信号の論理積または論理和を生成するゲートモジュールを選定し、このゲートモジュールに関する記述を生成して保持する。

#### 【 0 0 5 7 】

実際の記述では、特殊モジュールとして A N D ゲートモジュールを指定するには、記述生成部 1 2 は例えば以下のような固定モジュールのインスタンス呼出文を記述する。

```
AND N01(.A(分岐モジュールの通常バス N P 1 のノードの指定),.B(制御バス C
P 上の信号名の指定),.Y(セクタモジュール S L 5 の入力部に接続された通常
バス N P 2 のノードの指定));
```

ここで A N D N 0 1 とはゲートモジュールの呼び名の例であり、ゲートモジュールの呼び名はこの表記に限らず設計者が任意に決定できる。

特殊モジュールとして O R ゲートモジュールを指定するには、記述生成部 1 2

は以下のような固定モジュールのインスタンス呼出文を記述する。

OR N02(.A(分岐モジュールの通常パスNP1のノードの指定),.B(制御パスCP上の信号名の指定),.Y(セレクタモジュールSL5の入力部に接続された通常パスNP2のノードの指定));

ここでOR N02とはゲートモジュールの呼び名の例であり、ゲートモジュールの呼び名はこの表記に限らず設計者が任意に決定できる。

#### 【0058】

LSIの動作中に常に給電されている（給電の停止が不可能な）ブロック同士、または同じタイミングで給電が開始および停止されるブロック同士の接続では、かかる特殊モジュールは必要ない。しかしながら、二つのブロックの境界BDを横切る接続線に必ず特殊モジュールを設けるようにルール付けした場合には、本質的に必要でなくても形式的に特殊モジュールを設ける必要がある。そのような場合には、形式的に設けられた特殊モジュールに単にバッファまたは接続線を設ければよい。記述生成部12は、特殊モジュールとして、バッファのみを配置した定義文を生成するか、単純な接続定義文を生成する。

#### 【0059】

ステップST7（図2）の各IP接続の設定では、記述生成部12は、解析部11の解析結果に基づいて、LSI内の複数のIPに関する接続条件とこれらの接続条件に応じた各IPの接続に関する記述をHDLで生成する。従って、この実施の形態では、ピン（パッドまたは入出力セル）とIPとの間の接続を記述することができるだけでなく、ステップST7により複数のIPの間の接続を記述することが可能である。

#### 【0060】

従って、図14に示すように、全体のLSI内の複数の回路部品（ピンおよびIP）に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述したトップ階層の一覧表と、各IP内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の複数の一覧表とをステップST1で設計者が完成させていることが、各IP接続の設定の前提条件である。ステップST2の一覧表の読み込みでは、解析部11は、各ピンに

対応するピンオブジェクトに加えて、トップ階層の一覧表（L S I 全体）を表す I P オブジェクトと、下位階層の一覧表（各 I P）を表す複数の I P オブジェクトを保持しておく。ステップ S T 3 ～ S T 6 では、上記と同様に、オブジェクトで使用されたメソッドとしてのモジュールまたは信号の記述を追加または変更する。そして、ステップ S T 7 で、各 I P の接続に関する記述を生成して保持する。

#### 【 0 0 6 1 】

図 3 のように一枚の一覧表しか設計者が準備しない場合には、I P とパッドの接続のみしか一覧表に記述できなかったため、複数の I P 同士の接続を H D L で記述できないが、各 I P 内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の複数の一覧表を準備して、これらの一覧表に基づいて I P 間の接続を H D L で記述することが可能である。これらの一覧表の作成において、内部部品の接続が成立する条件を統一的な規則に従って表記しておけば、この自動回路設計装置でステップ S T 4 で作成する多数のセクタモジュールの制御信号を生成する制御信号生成モジュール（エンコーダモジュール）を共通に作成できる。すなわち、制御信号生成モジュールの個数を最小限にすることが可能である。

#### 【 0 0 6 2 】

図 1 5 は、複数の一覧表に基づいて各 I P 接続の設定を経て自動設計されうるトップ階層回路を示す回路図である。図 1 5 に示すように、ピン P A ～ P F がセクタモジュール S L 1 ～ S L 6 を介して I P A, B に接続されているだけでなく、I P B の出力ピンがセクタモジュール S L 2 を介して I P A の入力ピンに接続され、I P A の出力ピンがセクタモジュール S L 3, S L 4 を介して I P B の入力ピンに接続されている。このように I P 間の接続も存在するかなり複雑な回路を自動設計することができる。複数のセクタモジュール S L 1 ～ S L 6 の全てまたはいずれかは、共通の制御信号生成モジュールによって制御できるように制御信号生成モジュールを設計することができる。

#### 【 0 0 6 3 】

最後の工程としてステップ S T 8（図 2）の H D L による回路記述出力では、

今までのセクタモジュールの挿入、分岐モジュールの挿入、特殊モジュールの挿入、および各 I P の接続の各工程で保持（登録）された記述を、回路言語記述の文法に則って出力する。具体的には、設計対象の L S I 全体で使用される各モジュールを定義し、各モジュール定義の中で各 I P およびピン（パッドまたは入出力セル）の接続を示したインスタンス定義を記述する。また、登録された分岐モジュール、セクタモジュール、特殊モジュールのインスタンス定義を対象 L S I 全体のモジュール定義の中で行う。その後、対象 L S I 全体のモジュール定義をクローズし、セクタモジュールのモジュール定義を付加する。さらに、制御信号生成モジュール（エンコーダモジュール）のモジュール定義を付加する。このようにして設計対象の L S I 全体に関する H D L （R T L）による記述が出力され、R T L のファイルが作成される。

## 【 0 0 6 4 】

さらに、一覧表を利用して、論理合成ツールに適用する合成スクリプトを自動生成し、これらの合成スクリプトを用いて論理合成を行うようにプログラミングしてもよい。ステップ S T 8 で出力された L S I 全体に関する R T L による回路記述は論理合成することができ、これによりネットリストの論理回路が自動的に生成される。合成スクリプトは、R T L による回路記述をネットリストに論理合成するための設計制約条件または規則をスクリプト言語で記述した簡単なプログラムである。

## 【 0 0 6 5 】

プロセッサ 1 は、一覧表から論理合成するための合成条件または規則を抽出し、これらの合成条件または規則から合成スクリプトを作成することができる。この合成スクリプトを論理合成ツールに読み込ますだけで、論理合成ができるようになる。上述のように接続条件によって各回路部品の接続関係が異なるために、複数の論理回路を作成する必要がある、このため複数の合成スクリプトが必要である。R T L に指定するインスタンス文が多量にある場合に、多数の合成スクリプトを作成するのは手間がかかるが、自動的に合成スクリプトを生成することによって合成効率を高めることが可能である。

## 【 0 0 6 6 】



さらに一覧表を利用して、LSI内の各回路部品の接続を検証するためのテストパターンを作成し、これらのテストパターンを用いて接続を検証するようにプログラミングしてもよい。各テストパターンは、各IPの出力ノードに強制的かつ仮想的に印加され、その出力ノードの接続先の入力ノードから同じテストパターンが読み取られれば、接続が確認されたことになる。

## 【0067】

プロセッサ1は、一覧表から接続確認のためのテストパターンを自動生成できる。テストパターンは簡単ではあるが記述量が多く生成するためかなりの手間がかかるが、自動的にテストパターンを生成することによって検証効率を高めることが可能である。ここで行う検証は、HDLによる回路記述に対して行う検証でもよいし、合成されたネットリストに対して行う検証でもよい。

## 【0068】

以上のように、この実施の形態1によれば、プログラムによって動作する自動回路設計装置が、LSI内の複数の回路部品に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した一覧表を解析する解析部11と、解析部11の解析結果に基づいて、HDLによるLSIの記述を生成する記述生成部12とを備えるので、複数の回路部品を接続した集積回路の自動設計を容易にし、HDLによる回路記述の作成にかかる手間と時間を短縮することができるという効果がある。設計者が作成する一覧表は、HDLの文法に関する知識や回路に関する部品配置のノウハウを必要とせず、設計者は、接続が成立する条件とその条件に応じた接続関係のみを一覧表に規定すればよい。設計者は一覧表を単純なフォーマットに従って埋めていくだけでよい。また、一覧表のフォーマットは至って簡単であるため、HDLに比べて習得が容易であり、特別な技術を有さない設計者でも作成が可能となる。

## 【0069】

このような一覧表に基づいた回路設計は、複数のモジュールを接続する回路において有効なので、トップ階層回路の設計だけでなく、比較的上位階層の接続設計に便利である。

## 【0070】

特に、この実施の形態 1 では、解析部 1 1 の解析結果に基づいて、記述生成部 1 2 は、接続条件に応じた各回路部品の接続関係を実現するのに適切なセレクトラモジュールを選定し、L S I の記述にセレクトラモジュールに関する情報を付加するので、接続条件に応じてセレクトラモジュールを設計者自身が選定する必要がなく、特別な技術を有さない設計者でも H D L による回路設計が可能である。

## 【 0 0 7 1 】

また、この実施の形態 1 では、解析部 1 1 の解析結果に基づいて、記述生成部 1 2 は、セレクトラモジュールを制御する制御信号を生成するのに適切な制御信号生成モジュールに関する情報を L S I の記述に付加するので、接続条件に応じて制御信号生成モジュール（エンコーダモジュール）を設計者自身が選定する必要がなく、特別な技術を有さない設計者でも H D L による回路設計が可能である。

## 【 0 0 7 2 】

さらに、この実施の形態 1 では、L S I を通常使用する通常使用モードと L S I を副次的に使用する複数の副次的モードのそれぞれを接続条件とした一覧表を解析部 1 1 が解析し、解析部 1 1 の解析結果に基づいて、記述生成部 1 2 は、通常使用モードで接続される複数の回路部品の間に介在する第 1 のセレクトラモジュール S L 3 を選定し、複数の副次的モードでの各回路部品の接続関係を実現するのに適切な第 2 のセレクトラモジュール S L 4 を選定し、第 2 のセレクトラモジュール S L 4 の出力部が第 1 のセレクトラモジュール S L 3 の入力部に接続されるように、記述生成部 1 2 が L S I の記述に第 1 のセレクトラモジュール S L 3 と第 2 のセレクトラモジュール S L 4 に関する情報を付加するようにした。従って、設計された L S I においては、一つの通常パス N P と複数のテストパス T P が設けられても、第 1 のセレクトラモジュール S L 3 は、通常パス N P とテストパス群の二つの選択肢しか持たないので、その内部構成（例えば内部ゲートの個数や配線パターン）はセレクトラのうち最も簡単である。このため通常使用モードでは、高い処理速度と、低い電力消費を達成することが可能である。

## 【 0 0 7 3 】

さらに、この実施の形態 1 では、L S I を通常使用する通常使用モードと L S I を副次的に使用する副次的モードのそれぞれを接続条件とした一覧表を解析部

1 1 が解析し、解析部 1 1 の解析結果に基づいて、副次的モードで接続される複数の回路部品の中に副次的モードで信号を有効に伝達するゲートモジュール 3 4 を記述生成部 1 2 が選定し、記述生成部 1 2 が L S I の記述にゲートモジュール 3 4 に関する情報を付加する。従って、通常使用モードではゲートモジュール 3 4 の出力部の電位はローレベルに固定され、電力消費が抑えられる。

#### 【0074】

さらに、この実施の形態 1 では、接続条件と接続関係に加えて、各回路部品への給電の停止が可能か否かを記述したデータ集合を解析部 1 1 が解析し、解析部 1 1 の解析結果に基づいて、給電の停止が可能な回路部品と他の回路部品との間に配置されて、これらの回路部品間の信号の伝達を制御するゲートモジュール 4 4 を記述生成部 1 2 が選定し、記述生成部 1 2 が L S I の記述にゲートモジュール 4 4 に関する情報を付加する。従って、給電の停止が可能な回路部品への給電停止後に、ゲートモジュール 4 4 の出力部の電位はローレベルまたはハイレベルに固定され、不適切な信号が供給される事態を避けることが可能である。

#### 【0075】

さらに、この実施の形態 1 では、L S I 内の複数の回路部品に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した上位階層の一覧表と、各回路部品内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の複数の一覧表を解析部 1 1 が解析し、解析部 1 1 の解析結果に基づいて、記述生成部 1 2 は、L S I 全体および各回路部品に関する総合的な記述を一度に生成する。従って、これらの一覧表に基づいて I P 間の接続も存在するかなり複雑な回路を H D L で自動設計することが可能である。

#### 【0076】

##### 【実施例】

全体の L S I 内の接続関係を記述したトップ階層の一覧表と、各 I P 内の接続関係を記述した下位階層の複数の一覧表とを作成し、実際に H D L による L S I の記述を生成した。

図 1 6 はトップ階層の一覧表であり、図 1 7 は設計したトップ階層回路の通常

使用モードでの回路図であり、図 1 8 ～ 図 2 0 はそれぞれトップ階層回路内の I P A ～ C の一覧表である。図 1 7 に矩形の箱として示すように、トップ階層回路内には三つの I P A ～ C が存在する。

## 【 0 0 7 7 】

トップ階層回路の接続を示した図 1 6 の左端の列には L S I の端子である入出力バッファのピン A ～ H, T 1, T 2 の名前が記入されている。ピン A, E, D, T 1, T 2 は入力バッファの出力ピンでありその出力信号名が Y であり、ピン H は出力バッファの入力ピンでありその入力信号名が A であり、他のピン B, C, F, G は双方向ピンでありその入力または出力信号名が A, Y, C, I E であることが記入されている。

## 【 0 0 7 8 】

L S I の動作モード（接続条件）は 5 通りある。すなわち通常モード（通常使用モード）と各 I P A ～ C の切り出しモード（各 I P A ～ C を個別に試験するテストモード）であるが、I P C に関しては 2 通りの切り出しモードがある。一覧表中の各モードの名前の下には、そのモードの成立条件が記入されている。ここでは、パッドのピン T 1, T 2, C の信号内容によってモードが設定されること表されている。例えば、ピン T 1, T 2 上の信号が 0 なら通常モードであり、全てのピン T 1, T 2, C の信号が 1 なら C 切り出しモード 2 である。

## 【 0 0 7 9 】

図 1 6 では、各モードにおいて、ピン A ～ H, T 1, T 2 が、どの I P の入力ピンまたは出力ピンとしてのどのピンに接続されるかが記されている。一覧表中の I P の列は I P を識別し、P I N ( I P ) の列は I P 内のピンを識別し、I O の列は入力ピン ( I ) か出力ピン ( O ) かを識別する。例えば、図 1 6 に示すように、通常モードで、ピン A は I P A の入力ピンとしてのピン b に接続され、ピン B は I P A の入力ピンとしてのピン a に接続される。

## 【 0 0 8 0 】

図 1 8 ～ 図 2 0 では、それぞれ I P A ～ C 内のピン名と各モードでのピンの接続先が記入されており、図 1 6 と同様に各モードとモードの成立条件が記入されている。モードの記載事項に関しては全ての一覧表で統一されている。

## 【0081】

図18～図20の左端の列にはIPA～C内のピン名が記入され、次の列には各ピンが入力ピン（I）か出力ピン（O）か記入される。図18～図20では、各モードにおいて、各ピンが、どのIP（またはパッド）の入力ピンまたは出力ピンとしてどのピンに接続されるかが記されている。一覧表中のIPの列はIPまたはパッドを識別し、PIN（IP）の列はIPまたはパッド内のピンを識別し、IOの列は入力ピン（I）か出力ピン（O）かを識別する。

## 【0082】

これらの一覧表相互において、接続関係には矛盾がないように記入される。例えば、図16では、通常モードでピンAはIPAの入力ピンとしてのピンbに接続されることが示されているが、図18では、通常モードでIPAのピンbは、入力ピンとしてのパッドのピンAに接続されることが示されている。

## 【0083】

また、各一覧表には、トップ階層回路またはIPが接続される電源の種類と特性が示されている。図16および図20に示すように、トップ階層回路およびIPCの電源はAであり、これはLSIの動作中は常にオンである。図18および図19に示すように、IPA、Bの電源はBであり、これはオンにもオフにもなりうる。従って、IPCはLSIの動作中は常に給電されるが、IPA、BはLSIの動作中に給電の停止が可能である。

## 【0084】

これらの一覧表は、プログラムによって読み込まれ（図2のステップST2、ステップST7）、一覧表の各行の接続情報が、各ピンの接続先を格納したピンオブジェクトと、各一覧表ごとに設定され、ピンオブジェクトのデータを含むIPオブジェクトに格納される。ピンオブジェクトでは、各一覧表の左端のピンとその接続先を結ぶ信号線が指定されて登録される。また、各一覧表全体を表すIPオブジェクトに、各IPへの給電の停止が可能か否かを示す情報を保持しておく。

## 【0085】

最初に各モードの成立条件（接続条件）が評価される（図2のステップST3

）。このLSIの場合は、パッドの3本のピンT1, T2, Cの信号内容の組合せによってモードが決定される。一覧表に記載されたモードの成立条件（接続条件）に基づいて、例えば以下のようにモードが記述される。

```
function MODESE begin
```

```
    input [2:0] in;
```

```
    output [4:0] mode;
```

```
    casex (in) begin
```

```
        3'b00x: mode = 5'b00001;
```

```
        3'b01x: mode = 5'b00010;
```

```
        3'b10x: mode = 5'b00100;
```

```
        3'b110: mode = 5'b01000;
```

```
        3'b111: mode = 5'b10000;
```

```
    end
```

```
end
```

```
    assign {LSI_MODE_04, LSI_MODE_03, LSI_MODE_02, LSI_MODE_01, LSI_MODE_00} = MODESE({PAD_T1_X, PAD_T2_Y, PAD_C_Y});
```

【0086】

これにより、パッドのピンT1からの信号PAD\_T1\_X、ピンT2からの信号PAD\_T2\_Y、ピンCからの信号PAD\_C\_Yの組み合わせ[00x, 01x, 10x, 110, 111]（各一覧表参照）により、動作モードLSI\_MODE\_00からLSI\_MODE\_04が決定される。動作モードLSI\_MODE\_00は通常モード、動作モードLSI\_MODE\_04はC切り出しモード2である。

【0087】

その後、プロセッサ1は、各出力ピンについては、その出力ピンに対する接続先を調査して、接続認識配列を生成し、これに基づいて分岐モジュールを作成する（図2のステップST5）。また各入力ピンについては、その入力ピンに対する接続先を調査して、接続認識配列を生成し、これに基づいてセクタモジュールを生成する（図2のステップST4）。

【0088】

例えば、図 1 6 のピン A は、入力バッファの出力ピンなので出力信号 Y が各 I P に入力される。ピン A は、通常モードおよび A 切り出しテストモードでも I P A のピン b に接続されるが、B 切り出しモードでは I P B のピン a に接続され、C 切り出しモードでは I P C のピン a に接続される。従って、ピン A の接続先を表す 1, 1, 2, 3, 3 という接続認識配列が形成され、接続数は 3 となる。接続認識配列中の 1 は I P A のピン b、2 は I P B のピン a、3 は I P C のピン a を表す。接続数が 3 なので、分岐モジュールが形成され、各分岐モジュールの出力ノードが定義される。例えばピン A から I P A に向かう通常パスのノード名は PAD\_A\_N, ピン A から I P B と I P C に向かうテストパスのノード名は、PAD\_A\_M と定義される。ピン A の出力信号 Y は PAD\_A\_Y と定義されストックされる。

## 【 0 0 8 9 】

この情報は、HDL による回路記述出力（図 2 のステップ S T 8）で例えば以下のようなインスタンス呼出文として出力される。

```
TKM T01(.A(PAD_A_Y),.B(LSI_MODE_00_B),.N(PAD_A_N),.T(PAD_A_M));
```

ここで TKM T01 とはゲートモジュールの呼び名の例であり、ゲートモジュールの呼び名はこの表記に限らず設計者が任意に決定できる。その後のカッコ内の配列は、(.A(I P の出力ピンの出力信号名の指定),.B(通常モードでローレベルになる信号名の指定),.N(通常パスノードの指定),.T(テストパスノードの指定))である。例えば、LSI\_MODE\_00\_B は通常モード時にローレベル、それ以外でハイレベルとなる信号であり、エンコーダモジュール（ゲート制御モジュール）の挿入時に生成され定義される。

## 【 0 0 9 0 】

一方、図 2 0 に記された I P C のピン P C A のように、出力ピンでありながらも、接続先が全ての条件で同じ場合（例えば MODE モジュールのピン A to C に接続されるような場合）、接続数は 1 であり、接続認識配列が 1, 1, 1, 1, 1 となる。この場合には、分岐モジュールは生成されず、分岐モジュールの入るべきところには接続を示す記述、あるいは単なるバッファのみを示す文が挿入される。単なるバッファの場合は以下のように記述される。

```
TKB T13(.A(C_PCA_0),.Y(C_PCA_N));
```



ここでTKB T13とはバッファモジュールの呼び名の例であり、バッファモジュールの呼び名はこの表記に限らず設計者が任意に決定できる。その後のカッコ内の配列は、(.A(バッファの入力信号線の指定),.Y(バッファの出力信号線の指定))である。つまり、MODEモジュールに向かう信号線は、C\_PCA\_Nと定義され、記憶される。

#### 【0091】

入力ピンについては、複数の接続先がある場合には、セクタモジュールが挿入される（図2のステップST4）。この場合もまず接続認識配列が作成され、その接続認識配列に沿ったセクタモジュールが生成される。

#### 【0092】

例えば、図16のピンHは、出力バッファの入力ピンなので、セクタモジュールが生成される。ピンHは、通常モードではIPCのピンeに接続され、A切り出しモードではIPAのピンfに接続され、B切り出しモードではIPBのピンfに接続され、C切り出しモードでは、通常モードと同じようにIPCのピンeに接続される。この場合、接続認識配列は1, 2, 3, 1, 1となる。接続認識配列中の1はIPCのピンe、2はIPAのピンf、3はIPBのピンfを表す。接続数が3なので、通常パスとテストパスを分ける第1のセクタモジュールと、テストパス同士を分ける第2のセクタモジュールが生成される。第1のセクタモジュールの通常パスが接続される入力端子をNとすると、入力端子Nには、接続先の分岐モジュールの生成のときに通常パスとして生成されているはずであるC\_e\_Nという信号線が通常パスとして接続されるように登録される。また、第2のセクタモジュールの一方のテストパスが接続される入力端子には、A切り出しモードでの接続先であるA\_f\_Mという信号線がテストパスとして接続されるように登録される。信号線A\_f\_Mは、IPAの一覧表である図18のIPAのピンfの接続先として分岐モジュールが生成されたときに定義されているはずである。さらに、第2のセクタモジュールのもう一方のテストパスが接続される入力端子には、B切り出しモードでの接続先であるB\_f\_Mという信号線がテストパスとして接続されるように登録される。信号線B\_f\_Mは、IPBの一覧表である図19のIPBのピンfの接続先として分岐モジュールが生成されたときに



定義されているはずである。

【 0 0 9 3 】

さらに、セクタモジュールを制御するために、前述したモード条件（接続条件）の成立信号に応じて、エンコーダモジュールからセクタモジュール制御信号が各セクタモジュールに入力されるように設定する。セクタモジュール制御信号を入力するためのエンコーダモジュールの設定では、接続認識配列（例えば 1, 2, 3, 1, 1）に基づいて、通常モードの接続先が 1 ならば、図 9 の信号線 B A 上の信号 A が通常パスを選択するように、図 9 の信号線 B B 上の信号 B がいずれかのパスを選択するようにし、接続先が 2 なら信号線 B A 上の信号 A がテストパス、信号線 B B 上の信号 B がテストパス 1 を選択するようにし、接続先 3 なら、信号線 B A 上の信号 A がテストパス、信号線 B B 上の信号 B がテストパス 2 を選択するように設定する。このエンコーダモジュールは、HDL 記述の時には、各モード条件の成立信号を条件とした if 文の条件分岐式で例えば次のように定義されることになる。

【 0 0 9 4 】

```

if (P0==1'b1) begin
A=1'b0;  //接続先が1の場合のセクタモジュール制御信号パターン
B=1'b0;
end
else if (P1==1'b1) begin
A=1'b1;  //接続先が2の場合のセクタモジュール制御信号パターン
B=1'b0;
end
else if (P2==1'b1) begin
A=1'b1;  //接続先が3の場合のセクタモジュール制御信号パターン
B=1'b1;
end
else if (P3==1'b1) begin
A=1'b0;  //接続先が1の場合のセクタモジュール制御信号パターン

```

```

B=1'b0;
end
else if (P4==1'b1) begin
A=1'b0; //接続先が1の場合のセクタモジュール制御信号パターン
B=1'b0;
end

```

## 【 0 0 9 5 】

ここで、// はコメント開始記号であり、コメント開始記号から行末まではコメントである。P0からP4の信号は、モード条件成立信号（LSI\_MODE\_00からLSI\_MODE\_04）が入力されるピンのモジュール内部信号名である。上記の条件分岐式によれば、異なるモード条件成立信号について、複数のテストパスが同時に有効になることはない。

## 【 0 0 9 6 】

また、二つのセクタモジュール（第1および第2のセクタモジュール）は、例えば以下のようなモジュールのインスタンス呼出文として定義される。

```

TSEL2T01(.A(C_e_N),.B(SST),.S(A),.Y(PAD_H_A)); //通常パスとテストパス
TSEL2G01(.A(A_f_M),.B(B_f_M),.S(B),.Y(SST)); //テストパス1とテストパス2

```

ここで、TSEL2は二つのセクタモジュールの組み合わせのモジュールのモジュール名であり、T01は第1のセクタモジュール、G01は第2のセクタモジュールである。カッコ内のA,Bはセクタモジュールへの入力信号を示し、Sはセクタモジュール制御信号を示し、Yはセクタモジュールの出力信号を示す。SSTは、第2のセクタモジュールG01から出力され、第1のセクタモジュールT01へ入力される信号である。

## 【 0 0 9 7 】

セクタモジュールは、複数の接続先がある入力ピンについて必要となるのであり、接続認識配列が数字1のみから構成される（接続数が1）入力ピンには、セクタモジュールが挿入されず、接続を示す定義のみのモジュールか、あるいはバッファのみのモジュールがセクタモジュールの位置に挿入される。例えば

、図 2 0 のピン test のように入力ピンでありながらも、接続先が全ての条件で同じ場合（例えば MODE モジュールのピン CT に接続されるような場合）、接続数は 1 であり、接続認識配列が 1, 1, 1, 1, 1 となる。この場合には、セレクトタモジュールは生成されず、セレクトタモジュールの入るべきところには接続を示す記述、あるいは単なるバッファのみを示す文が挿入される。接続を示す記述は例えば以下の通りである。

【 0 0 9 8 】

```
CONNE01 T02(.A(MODE_cT_N),.Y(C_test_I));
```

ここで CONNE01 は接続のみを示すモジュールであり、カッコ内の A, Y は互いに接続されるピンを示す。

【 0 0 9 9 】

各セレクトタモジュールの内部回路は、接続先の個数に応じて一意に決定される。従って、設計される L S I で使用されるセレクトタモジュールの種類は、設計される L S I にありうる接続数の種類の分だけ存在することになる。各セレクトタモジュールが登録されるときに接続先の個数が調べられ、新規な個数であれば（調べられた個数が既に使用されたセレクトタモジュールの接続先の個数のどれにも一致しなければ）、新規な種類のセレクトタモジュールの定義を記述して、これを登録する。過去に接続先の個数が同じセレクトタモジュールの定義が登録されていれば、セレクトタモジュールの種類はしないようにして、重複した定義をさけ、インスタンス呼出文のみを生成して登録する。

【 0 1 0 0 】

また、複数のセレクトタモジュールの組合せからなるモジュール全体の構成は、接続認識配列の種類に応じて異なる。同じ I P の複数の入力ピン同士のように、接続認識配列が同じ複数の入力ピンは存在しうる。従って、接続認識配列を登録しておき、過去に同じ接続認識配列をもつモジュールが定義されていれば、重複を避けるために、モジュール定義をせず、インスタンス呼出文のみ定義するとよい。新規の接続認識配列の場合は、モジュール定義を記述してこれを登録する。

【 0 1 0 1 】

上述のように各一覧表には、トップ階層回路または I P が接続される電源の種

類と特性が示されている。L S I の動作中は常に給電されているブロックと、給電の停止が可能なブロックが接続される場合、特殊モジュールが挿入される（図 2 のステップ S T 6）。これは給電の停止が可能なブロックの電源がオフしてそのブロックの出力が中間電位になる場合に、常に給電されているブロックのゲートが中間電位のリーク電流を流す可能性を排除するためである。このため、ブロックの相互の間に A N D ゲートモジュールを挿入し、給電の停止が可能なブロックの給電が停止になった場合、この A N D ゲートモジュールの一方の入力部にローレベルの信号を入れて、このブロックの出力電位をローレベルに固定する。

#### 【 0 1 0 2 】

図 1 6 および図 2 0 に示すように、トップ階層回路および I P C の電源は A であり、これは L S I の動作中は常にオンである。図 1 8 および図 1 9 に示すように、I P A, B の電源は B であり、これはオンにもオフにもなりうる。従って、I P A の出力ピンと他の I P またはパッドとの間には特殊モジュールが挿入され、I P B の出力ピンと他の I P またはパッドとの間にも特殊モジュールが挿入される。他の I P または入力パッドの出力ピンには、特殊モジュールは挿入されず、接続を示す記述、あるいは単なるバッファのみを示す文が挿入される。特殊モジュールとしてのゲートモジュールは上述の通りインスタンス呼出文で記述される。

#### 【 0 1 0 3 】

最後の工程としてステップ S T 8（図 2）の H D L による回路記述出力では、今までの各工程で保持（登録）された記述を R T L の文法に則って出力した。さらに、上記の通り、合成スクリプトを自動生成し、これらの合成スクリプトを用いて論理合成した。また、上記の通り、合成されたネットリストに対して、接続を検証するためのテストパターンを作成し、自動検証を行った。

#### 【 0 1 0 4 】

以下は、生成されたトップ階層回路を表す R T L 記述である。

```
module TOP (A,D,E,T1,T2,
            H,J,
            B,C,F,G);
```

```

input    A,D,E,T1,T2;
output   H,J;
inout    B,C,F,G;
wire      PAD_A_Y,PAD_A_M,PAD_A_N,PAD_B_A,PAD_B_Y,PAD_B_M,PAD_
B_C,
          PAD_B_IE,PAD_C_A,PAD_C_Y,PAD_C_M,PAD_C_N,PAD_C_C,PAD_C_IE,
          PAD_D_Y,PAD_D_M,PAD_D_N,PAD_E_Y,PAD_E_M,PAD_E_N,PAD_F_A,
          PAD_F_Y,PAD_F_M,PAD_F_C,PAD_F_IE,PAD_G_A,PAD_G_Y,PAD_G_M,
          PAD_G_N,PAD_G_C,PAD_G_IE,PAD_J_A,PAD_J_C,PAD_H_A,PAD_T1_Y,
          PAD_T1_M,PAD_T2_Y,PAD_T2_M,
          MODE_t1_I,MODE_t2_I,MODE_cT_O,MODE_subT_I,
          A_a_I,A_b_I,A_c_I,A_d_O,A_d_M,A_d_N,A_e_O,A_e_M,A_e_N,
          A_f_O,A_f_M,A_f_N,A_g_I,A_FCL_O,A_FCL_M,A_JA_O,A_JA_M,
          A_JC_O,A_JC_M,A_h_I,
          B_e_O,B_e_M,B_d_O,B_d_M,B_a_I,B_b_I,B_c_I,B_f_O,B_f_M,
          B_f_N,
          C_d_I,C_c_I,C_e_O,C_e_M,C_a_I,C_b_O,C_b_M,C_b_N,C_f_I,
          C_g_O,C_g_M,C_g_N,C_test_I,C_PCA_O,C_PCA_M,C_PCB_O,
          C_PCB_M,
          LSI_TEST_MODE00,LSI_TEST_MODE01,LSI_TEST_MODE02,LSI_TEST_MODE03,
          LSI_TEST_MODE04,
          L_L_L,H_H_H,LSI_TEST_MODE00_B;

assign L_L_L = 1'b0;
assign H_H_H = 1'b1;

ICIUD1011    BUF_A (.Y(PAD_A_Y),.PAD(A));
BCTNEUC204SA21  BUF_B (.IE(PAD_B_IE),.Y(PAD_B_Y),.A(PAD_B_A),.C(PA

```

D\_B\_C),.PAD(B));

BCTNEUC204SA21 BUF\_C (.IE(PAD\_C\_IE),.Y(PAD\_C\_Y),.A(PAD\_C\_A),.C(PAD\_C\_C),.PAD(C));

ICN21 BUF\_D (.Y(PAD\_D\_Y),.PAD(D));

ICN21 BUF\_E (.Y(PAD\_E\_Y),.PAD(E));

BCTNEDC204SA21 BUF\_F (.IE(PAD\_F\_IE),.Y(PAD\_F\_Y),.A(PAD\_F\_A),.C(PAD\_F\_C),.PAD(F));

BCTNEDC204SA21 BUF\_G (.IE(PAD\_G\_IE),.Y(PAD\_G\_Y),.A(PAD\_G\_A),.C(PAD\_G\_C),.PAD(G));

OCN2SA21 BUF\_H (.A(PAD\_H\_A),.PAD(H));

OCN2SA21 BUF\_J (.A(PAD\_J\_A),.C(PAD\_J\_C),.PAD(J));

ICN21 BUF\_T1 (.Y(PAD\_T1\_Y),.PAD(T1));

ICN21 BUF\_T2 (.Y(PAD\_T2\_Y),.PAD(T2));

LSI\_TEST\_DEC ULTDEC (.t1(MODE\_t1\_I),.t2(MODE\_t2\_I),.subT(MODE\_sub T\_I),

.M00(LSI\_TEST\_MODE00),.M01(LSI\_TEST\_MODE01),.M02(LSI\_TEST\_MODE02),.M03(LSI\_TEST\_MODE03),.M04(LSI\_TEST\_MODE04),.M0B(LSI\_TEST\_MODE00\_B),.cT(MODE\_cT\_O));

SELECTOR000 UTS0000 (.Y(PAD\_B\_A),.A00(L\_L\_L),.A01(C\_b\_N),.M00(LSI\_TEST\_MODE00),.M01(LSI\_TEST\_MODE01),.M02(LSI\_TEST\_MODE02)

,.M03(LSI\_TEST\_MODE03),.M04(LSI\_TEST\_MODE04));

SELECTOR000 UTS0001 (.Y(PAD\_B\_C),.A00(H\_H\_H),.A01(L\_L\_L),.M00(LSI\_TEST\_MODE00)

,.M01(LSI\_TEST\_MODE01),.M02(LSI\_TEST\_MODE02),.M03(LSI\_TEST\_MODE03),.M04(LSI\_TEST\_MODE04)

);

SELECTOR000 UTS0002 (.Y(PAD\_B\_IE),.A00(H\_H\_H),.A01(L\_L\_L),.M00(LSI\_TEST\_MODE00),.M01(LSI\_TEST\_MODE01),.M02(LSI\_TEST\_MODE02)

```

        ,.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE04));
SELECTOR001 UTS0003 (.Y(PAD_C_A),.A00(L_L_L),.M00(LSI_TEST_MODE00),.
M01(LSI_TEST_MODE01)
        ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR002 UTS0004 (.Y(PAD_C_C),.A00(H_H_H),.A01(L_L_L)
        ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
        ,.M04(LSI_TEST_MODE04));
SELECTOR002 UTS0005 (.Y(PAD_C_IE),.A00(H_H_H),.A01(L_L_L),.M00(LSI_T
EST_MODE00),.M01(LSI_TEST_MODE01)
        ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR003 UTS0006 (.Y(PAD_F_A),.A00(B_e_M),.A01(A_d_N)
        ,.A02(C_g_N),.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LS
I_TEST_MODE02),.M03(LSI_TEST_MODE03)
        ,.M04(LSI_TEST_MODE04));
SELECTOR004 UTS0007 (.Y(PAD_F_C),.A00(A_FCL_M),.A01(L_L_L),.M00(LSI_
TEST_MODE00),.M01(LSI_TEST_MODE01)
        ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR004 UTS0008 (.Y(PAD_F_IE),.A00(A_FCL_M),.A01(L_L_L)
        ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
        ,.M04(LSI_TEST_MODE04));
SELECTOR003 UTS0009 (.Y(PAD_G_A),.A00(B_d_M),.A01(A_e_N),.A02(L_L_L)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01)
        ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));

```

```

SELECTOR000 UTS0010 (.Y(PAD_G_C),.A00(L_L_L),.A01(H_H_H)
    ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));

SELECTOR000 UTS0011 (.Y(PAD_G_IE),.A00(L_L_L),.A01(H_H_H),.M00(LSI_T
EST_MODE00),.M01(LSI_TEST_MODE01)
    ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));

SELECTOR004 UTS0012 (.Y(PAD_J_A),.A00(A_JA_M),.A01(L_L_L)
    ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));

SELECTOR004 UTS0013 (.Y(PAD_J_C),.A00(A_JC_M),.A01(L_L_L),.M00(LSI_T
EST_MODE00),.M01(LSI_TEST_MODE01)
    ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));

SELECTOR005 UTS0014 (.Y(PAD_H_A),.A00(C_e_M),.A01(A_f_N)
    ,.A02(B_f_N),.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LS
I_TEST_MODE02),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));

NKOutDum UKB000 (.A(PAD_A_Y),.Y(PAD_A_M),.H(PAD_A_N),.TONC(LSI_TEST_
MODE00_B));

KOutDum UKB001 (.A(PAD_B_Y),.Y(PAD_B_M));

NKOutDum UKB002 (.A(PAD_C_Y),.Y(PAD_C_M),.H(PAD_C_N),.TONC(LSI_TEST_
MODE00_B));

NKOutDum UKB003 (.A(PAD_D_Y),.Y(PAD_D_M),.H(PAD_D_N),.TONC(LSI_TEST_
MODE00_B));

NKOutDum UKB004 (.A(PAD_E_Y),.Y(PAD_E_M),.H(PAD_E_N),.TONC(LSI_TEST_
MODE00_B));

```



```

KOutDum UKB005 (.A(PAD_F_Y),.Y(PAD_F_M));
NKOutDum UKB006 (.A(PAD_G_Y),.Y(PAD_G_M),.H(PAD_G_N),.TONC(LSI_TEST_
MODE00_B));
KOutDum UKB007 (.A(PAD_T1_Y),.Y(PAD_T1_M));
KOutDum UKB008 (.A(PAD_T2_Y),.Y(PAD_T2_M));
SELECTOR001 UTS0015 (.Y(MODE_t1_I),.A00(PAD_T1_M),.M00(LSI_TEST_MODE
00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE02)
,.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE04));
SELECTOR001 UTS0016 (.Y(MODE_t2_I),.A00(PAD_T2_M),.M00(LSI_TEST_MODE
00),.M01(LSI_TEST_MODE01)
,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR000 UTS0017 (.Y(MODE_subT_I),.A00(L_L_L),.A01(PAD_C_N)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
,.M04(LSI_TEST_MODE04));
KOutDum UKB009 (.A(MODE_cT_O),.Y(MODE_cT_M));
SELECTOR002 UTS0018 (.Y(A_a_I),.A00(PAD_B_M),.A01(L_L_L),.M00(LSI_TE
ST_MODE00),.M01(LSI_TEST_MODE01)
,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR002 UTS0019 (.Y(A_b_I),.A00(PAD_A_M),.A01(L_L_L)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
,.M04(LSI_TEST_MODE04));
SELECTOR002 UTS0020 (.Y(A_c_I),.A00(PAD_C_M),.A01(L_L_L),.M00(LSI_TE
ST_MODE00),.M01(LSI_TEST_MODE01)
,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));

```

```

SELECTOR006 UTS0021 (.Y(A_g_I),.A00(C_b_M),.A01(PAD_D_N)
    ,.A02(L_L_L),.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LS
I_TEST_MODE02),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));
SELECTOR002 UTS0022 (.Y(A_h_I),.A00(PAD_E_M),.A01(L_L_L),.M00(LSI_TE
ST_MODE00),.M01(LSI_TEST_MODE01)
    ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
NKOutDum UKB010 (.A(A_d_O),.Y(A_d_M),.H(A_d_N),.TONC(LSI_TEST_MODE00
_B));
NKOutDum UKB011 (.A(A_e_O),.Y(A_e_M),.H(A_e_N),.TONC(LSI_TEST_MODE00
_B));
NKOutDum UKB012 (.A(A_f_O),.Y(A_f_M),.H(A_f_N),.TONC(LSI_TEST_MODE00
_B));
KOutDum UKB013 (.A(A_FCL_O),.Y(A_FCL_M));
KOutDum UKB014 (.A(A_JA_O),.Y(A_JA_M));
KOutDum UKB015 (.A(A_JC_O),.Y(A_JC_M));
SELECTOR007 UTS0023 (.Y(B_a_I),.A00(A_d_M),.A01(L_L_L)
    ,.A02(PAD_A_N),.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(
LSI_TEST_MODE02),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));
SELECTOR007 UTS0024 (.Y(B_b_I),.A00(A_e_M),.A01(L_L_L),.A02(PAD_E_N)
    ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01)
    ,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR007 UTS0025 (.Y(B_c_I),.A00(C_g_M),.A02(PAD_D_N)
    ,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
    ,.M04(LSI_TEST_MODE04));

```

```

KOutDum UKB016 (.A(B_e_O),.Y(B_e_M));
KOutDum UKB017 (.A(B_d_O),.Y(B_d_M));
NKOutDum UKB018 (.A(B_f_O),.Y(B_f_M),.H(B_f_N),.TONC(LSI_TEST_MODE00
_B));
SELECTOR008 UTS0026 (.Y(C_d_I),.A00(PAD_E_M),.A01(L_L_L),.M00(LSI_TE
ST_MODE00),.M01(LSI_TEST_MODE01)
,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR008 UTS0027 (.Y(C_c_I),.A00(PAD_D_M),.A01(L_L_L)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
,.M04(LSI_TEST_MODE04));
SELECTOR009 UTS0028 (.Y(C_a_I),.A00(A_f_M),.A01(H_H_H),.A02(PAD_A_N)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01)
,.M02(LSI_TEST_MODE02),.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE0
4));
SELECTOR009 UTS0029 (.Y(C_f_I),.A00(B_f_M),.A02(PAD_G_N)
,.M00(LSI_TEST_MODE00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE0
2),.M03(LSI_TEST_MODE03)
,.M04(LSI_TEST_MODE04));
SELECTOR001 UTS0030 (.Y(C_test_I),.A00(MODE_cT_M),.M00(LSI_TEST_MODE
00),.M01(LSI_TEST_MODE01),.M02(LSI_TEST_MODE02)
,.M03(LSI_TEST_MODE03),.M04(LSI_TEST_MODE04));
KOutDum UKB019 (.A(C_e_O),.Y(C_e_M));
NKOutDum UKB020 (.A(C_b_O),.Y(C_b_M),.H(C_b_N),.TONC(LSI_TEST_MODE00
_B));
NKOutDum UKB021 (.A(C_g_O),.Y(C_g_M),.H(C_g_N),.TONC(LSI_TEST_MODE00
_B));
KOutDum UKB022 (.A(C_PCA_O),.Y(C_PCA_M));

```

```

KOutDum UKB023 (.A(C_PCB_O),.Y(C_PCB_M));
A UTOPM_A(.a(A_a_I),.b(A_b_I),.c(A_c_I),.g(A_g_I),.h(A_h_I),.d(A_d_O
),
    .e(A_e_O),.f(A_f_O),.FCL(A_FCL_O),.JA(A_JA_O),.JC(A_JC_O));
B UTOPM_B(.a(B_a_I),.b(B_b_I),.c(B_c_I),.e(B_e_O),.d(B_d_O),.f(B_f_O
));
C UTOPM_C(.d(C_d_I),.c(C_c_I),.a(C_a_I),.f(C_f_I),.test(C_test_I),
    .e(C_e_O),.b(C_b_O),.g(C_g_O),.PCA(C_PCA_O),.PCB(C_PCB_O));
KubiMoto    UKM00 (.M(LSI_TEST_MODE00_B),.K(C_PCA_M),.Q(C_PCA_Q),.B(
C_PCA_Q_b));
KubiMoto    UKM01 (.M(LSI_TEST_MODE00_B),.K(C_PCB_M),.Q(C_PCB_Q),.B(
C_PCB_Q_b));
endmodule

```

```

module LSI_TEST_DEC (t1,t2,subT,
    M00,M01,M02,M03,M04,M0B,
    cT);
    input    t1,t2,subT;
    output    M00,M01,M02,M03,M04,M0B;
    output    cT;

    wire      [2:0] MPW;
    wire      [4:0] MMW;

    assign    MPW[0] = t1;
    assign    MPW[1] = t2;
    assign    MPW[2] = subT;
    assign    M0B = ~M00;

```

```
assign    M00 = MMW[0];
assign    M01 = MMW[1];
assign    M02 = MMW[2];
assign    M03 = MMW[3];
assign    M04 = MMW[4];
```

```
function [4:0] MDec;
    input [2:0] GM;
    casex (GM)
        3'bx00:    MDec = 5'b00001;
        3'bx10:    MDec = 5'b00010;
        3'bx01:    MDec = 5'b00100;
        3'b011:    MDec = 5'b01000;
        3'b111:    MDec = 5'b10000;
        default:   MDec = 5'b00000;
    endcase
endfunction
```

```
assign    MMW = MDec(MPW);
```

```
function TestMode00;
    input [2:0] GW;
    casex (GW)
        3'bx00:    TestMode00 = 1'b0;
        3'bx10:    TestMode00 = 1'b0;
        3'bx01:    TestMode00 = 1'b0;
        3'b011:    TestMode00 = 1'b0;
        3'b111:    TestMode00 = 1'b1;
        default:   TestMode00 = 1'bx;
```

```

        endcase

    endfunction

    assign cT = TestMode00(MPW);
endmodule

module Sel1to1 (Y,A00,S0);

    input    A00,S0;
    output    Y;

    function sel;
        input    A00,WS;
        case (WS)
            0:    sel = A00;
            default:    sel = 1'b0;
        endcase
    endfunction

    assign Y = sel(A00,S0);
endmodule

module Enc0 (M00,M01,M02,M03,M04,S0,ST);

    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00,M01,M02;
        if ( M00 == 1'b1 | M01 == 1'b1 | M02 == 1'b1)

```

```

        Enc1 = 1'b1;
    else
        Enc1 = 1'b0;
    endfunction

function Enc2;
    input    M03,M04;
    if ( M03 == 1'b1 )
        Enc2 = 1'd0;
    else if ( M04 == 1'b1 )
        Enc2 = 1'd0;
    else
        Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00,M01,M02);
    assign    S0 = Enc2(M03,M04);
endmodule

module SELECTOR000 (Y,A00,A01,M00,M01,M02,M03,M04);
    input    A00,A01,M00,M01,M02,M03,M04;
    output    Y;
    wire    ST,SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(A01));
    Enc0 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

```

```
module SELECTOR001 (Y,A00,M00,M01,M02,M03,M04);
```

```
    input    A00,M00,M01,M02,M03,M04;
```

```
    output   Y;
```

```
        assign    Y = A00;
```

```
endmodule
```

```
module Enc2 (M00,M01,M02,M03,M04,S0,ST);
```

```
    input    M00,M01,M02,M03,M04;
```

```
    output   S0,ST;
```

```
function Enc1;
```

```
    input    M00,M01;
```

```
        if ( M00 == 1'b1 | M01 == 1'b1)
```

```
            Enc1 = 1'b1;
```

```
        else
```

```
            Enc1 = 1'b0;
```

```
endfunction
```

```
function Enc2;
```

```
    input    M02,M03,M04;
```

```
        if ( M02 == 1'b1 )
```

```
            Enc2 = 1'd0;
```

```
        else if ( M03 == 1'b1 )
```

```
            Enc2 = 1'd0;
```

```
        else if ( M04 == 1'b1 )
```

```
            Enc2 = 1'd0;
```

```
        else
```

```
            Enc2 = 1'd0;
```



endfunction

assign ST = Enc1(M00,M01);

assign S0 = Enc2(M02,M03,M04);

endmodule

module SELECTOR002 (Y,A00,A01,M00,M01,M02,M03,M04);

input A00,A01,M00,M01,M02,M03,M04;

output Y;

wire ST,SS;

TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(A01));

Enc2 US3 (.ST(ST),.S0(SS),

.M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));

endmodule

module Sel2to1 (Y,A00,A01,S0);

input A00,A01,S0;

output Y;

function sel;

input A00,A01,WS;

case (WS)

0: sel = A00;

1: sel = A01;

default: sel = 1'b0;

endcase

endfunction

```
    assign Y = sel(A00,A01,S0);
endmodule

module Enc3 (M00,M01,M02,M03,M04,S0,ST);
    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00,M02;
        if ( M00 == 1'b1 | M02 == 1'b1)
            Enc1 = 1'b1;
        else
            Enc1 = 1'b0;
    endfunction

    function Enc2;
        input    M01,M03,M04;
        if ( M01 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M03 == 1'b1 )
            Enc2 = 1'd1;
        else if ( M04 == 1'b1 )
            Enc2 = 1'd1;
        else
            Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00,M02);
```

```

        assign    S0 = Enc2(M01,M03,M04);
endmodule

module SELECTOR003 (Y,A00,A01,A02,M00,M01,M02,M03,M04);
    input    A00,A01,A02,M00,M01,M02,M03,M04;
    output    Y;

    wire    ST,S0;
    wire    SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(S0));
    Sel2to1 US2 (.Y(S0),
        .A00(A01),.A01(A02),
        .S0(SS));
    Enc3 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

module Enc4 (M00,M01,M02,M03,M04,S0,ST);
    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00;
        if ( M00 == 1'b1)
            Enc1 = 1'b1;
        else
            Enc1 = 1'b0;
    endfunction
endmodule

```

```

function Enc2;
    input    M01,M02,M03,M04;
        if ( M01 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M02 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M03 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M04 == 1'b1 )
            Enc2 = 1'd0;
        else
            Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00);
    assign    S0 = Enc2(M01,M02,M03,M04);
endmodule

module SELECTOR004 (Y,A00,A01,M00,M01,M02,M03,M04);
    input    A00,A01,M00,M01,M02,M03,M04;
    output    Y;
    wire    ST,SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(A01));
    Enc4 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

```

```

module Enc5 (M00,M01,M02,M03,M04,S0,ST);
    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00,M03,M04;
        if ( M00 == 1'b1 | M03 == 1'b1 | M04 == 1'b1)
            Enc1 = 1'b1;
        else
            Enc1 = 1'b0;
    endfunction

    function Enc2;
        input    M01,M02;
        if ( M01 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M02 == 1'b1 )
            Enc2 = 1'd1;
        else
            Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00,M03,M04);
    assign    S0 = Enc2(M01,M02);
endmodule

```

```

module SELECTOR005 (Y,A00,A01,A02,M00,M01,M02,M03,M04);
    input    A00,A01,A02,M00,M01,M02,M03,M04;
    output    Y;

```

```

wire    ST,S0;
wire    SS;

TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(S0));
Sel2to1 US2 (.Y(S0),
             .A00(A01),.A01(A02),
             .S0(SS));
Enc5 US3 (.ST(ST),.S0(SS),
          .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

module Enc6 (M00,M01,M02,M03,M04,S0,ST);
  input  M00,M01,M02,M03,M04;
  output S0,ST;

  function Enc1;
    input  M00;
    if ( M00 == 1'b1)
      Enc1 = 1'b1;
    else
      Enc1 = 1'b0;
  endfunction

  function Enc2;
    input  M01,M02,M03,M04;
    if ( M01 == 1'b1 )
      Enc2 = 1'd0;
    else if ( M02 == 1'b1 )

```

```

        Enc2 = 1'd1;
    else if ( M03 == 1'b1 )
        Enc2 = 1'd1;
    else if ( M04 == 1'b1 )
        Enc2 = 1'd1;
    else
        Enc2 = 1'd0;
endfunction

    assign    ST = Enc1(M00);
    assign    S0 = Enc2(M01,M02,M03,M04);
endmodule

module SELECTOR006 (Y,A00,A01,A02,M00,M01,M02,M03,M04);
    input    A00,A01,A02,M00,M01,M02,M03,M04;
    output   Y;

    wire     ST,S0;
    wire     SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(S0));
    Sel2to1 US2 (.Y(S0),
        .A00(A01),.A01(A02),
        .S0(SS));
    Enc6 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

module Enc7 (M00,M01,M02,M03,M04,S0,ST);

```

input M00,M01,M02,M03,M04;

output S0,ST;

function Enc1;

input M00;

if ( M00 == 1'b1)

Enc1 = 1'b1;

else

Enc1 = 1'b0;

endfunction

function Enc2;

input M01,M02,M03,M04;

if ( M01 == 1'b1 )

Enc2 = 1'd0;

else if ( M02 == 1'b1 )

Enc2 = 1'd1;

else if ( M03 == 1'b1 )

Enc2 = 1'd0;

else if ( M04 == 1'b1 )

Enc2 = 1'd0;

else

Enc2 = 1'd0;

endfunction

assign ST = Enc1(M00);

assign S0 = Enc2(M01,M02,M03,M04);

endmodule



```

module SELECTOR007 (Y,A00,A01,A02,M00,M01,M02,M03,M04);
    input    A00,A01,A02,M00,M01,M02,M03,M04;
    output    Y;

    wire    ST,S0;
    wire    SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(S0));
    Sel2to1 US2 (.Y(S0),
        .A00(A01),.A01(A02),
        .S0(SS));
    Enc7 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

```

```

module Enc8 (M00,M01,M02,M03,M04,S0,ST);
    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00,M03,M04;
        if ( M00 == 1'b1 | M03 == 1'b1 | M04 == 1'b1)
            Enc1 = 1'b1;
        else
            Enc1 = 1'b0;
    endfunction

```

```

function Enc2;
    input    M01,M02;

```

```

        if ( M01 == 1'b1 )
            Enc2 = 1'd0;
        else if ( M02 == 1'b1 )
            Enc2 = 1'd0;
        else
            Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00,M03,M04);
    assign    S0 = Enc2(M01,M02);
endmodule

module SELECTOR008 (Y,A00,A01,M00,M01,M02,M03,M04);
    input    A00,A01,M00,M01,M02,M03,M04;
    output    Y;
    wire    ST,SS;

    TSel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(A01));
    Enc8 US3 (.ST(ST),.S0(SS),
        .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));
endmodule

module Enc9 (M00,M01,M02,M03,M04,S0,ST);
    input    M00,M01,M02,M03,M04;
    output    S0,ST;

    function Enc1;
        input    M00;
        if ( M00 == 1'b1)

```

```

        Enc1 = 1'b1;
    else
        Enc1 = 1'b0;
    endfunction

function Enc2;
    input    M01,M02,M03,M04;
    if ( M01 == 1'b1 )
        Enc2 = 1'd0;
    else if ( M02 == 1'b1 )
        Enc2 = 1'd0;
    else if ( M03 == 1'b1 )
        Enc2 = 1'd1;
    else if ( M04 == 1'b1 )
        Enc2 = 1'd1;
    else
        Enc2 = 1'd0;
    endfunction

    assign    ST = Enc1(M00);
    assign    S0 = Enc2(M01,M02,M03,M04);
endmodule

module SELECTOR009 (Y,A00,A01,A02,M00,M01,M02,M03,M04);
    input    A00,A01,A02,M00,M01,M02,M03,M04;
    output    Y;

    wire    ST,S0;
    wire    SS;

```

```

Tsel2to1 US1 (.Y(Y),.S(ST),.A(A00),.B(S0));
Sel2to1 US2 (.Y(S0),
              .A00(A01),.A01(A02),
              .S0(SS));
Enc9 US3 (.ST(ST),.S0(SS),
           .M00(M00),.M01(M01),.M02(M02),.M03(M03),.M04(M04));

endmodule

```

【 0 1 0 5 】

#### 【発明の効果】

以上説明したように、この発明によれば、集積回路内の複数の回路部品に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した表形式のデータ集合を解析する解析部と、解析部の解析結果に基づいて、ハードウェア記述言語による集積回路の記述を生成する記述生成部とを備えることにより、複数の回路部品を接続した集積回路の自動設計を容易にし、ハードウェア記述言語による回路記述の作成にかかる手間と時間を短縮することが可能である。

#### 【図面の簡単な説明】

【図 1】 この発明の実施の形態 1 に係る自動回路設計装置を示すブロック図である。

【図 2】 プログラムに従って実施の形態 1 に係る自動回路設計装置が実行する動作を示すフローチャートである。

【図 3】 実施の形態 1 で使用されうる設計者に記入された一覧表の例である。

【図 4】 図 3 の一覧表に基づいて最終的に設計される目的の回路を示す回路図である。

【図 5】 図 3 の一覧表に基づいて作成された回路記述の一部を抜粋して示すリストである。

【図 6】 実施の形態 1 に従って L S I の記述を生成する工程の基本的考え方を説明するための図である。

【図 7】 図 3 の一覧表の接続条件を規定する条件定義文を平易化して示すリストである。

【図 8】 実施の形態 1 で使用されうる一覧表の例である。

【図 9】 図 8 に例示する一覧表に基づいて最終的に設計される目的の回路を示す回路図である。

【図 1 0】 図 9 に記載の回路に関する各セクタモジュールの記述を平易化して示すリストである。

【図 1 1】 図 9 に記載の回路に関するエンコーダモジュールの記述を平易化して示すリストである。

【図 1 2】 実施の形態 1 の動作により挿入された分岐モジュールを示す回路図である。

【図 1 3】 実施の形態 1 の動作の特殊モジュールの挿入で付加されたゲートモジュールを示す回路図である。

【図 1 4】 実施の形態 1 で使用されうる複数の一覧表の例である。

【図 1 5】 複数の一覧表に基づいて自動設計されうるトップ階層回路を示す回路図である。

【図 1 6】 実施例で作成した、全体の L S I 内の複数の回路部品（ピンおよび I P）に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述したトップ階層の一覧表である。

【図 1 7】 実施例で設計したトップ階層回路の回路図である。

【図 1 8】 実施例で作成した、一つの I P A 内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の一覧表である。

【図 1 9】 実施例で作成した、他の I P B 内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の一覧表である。

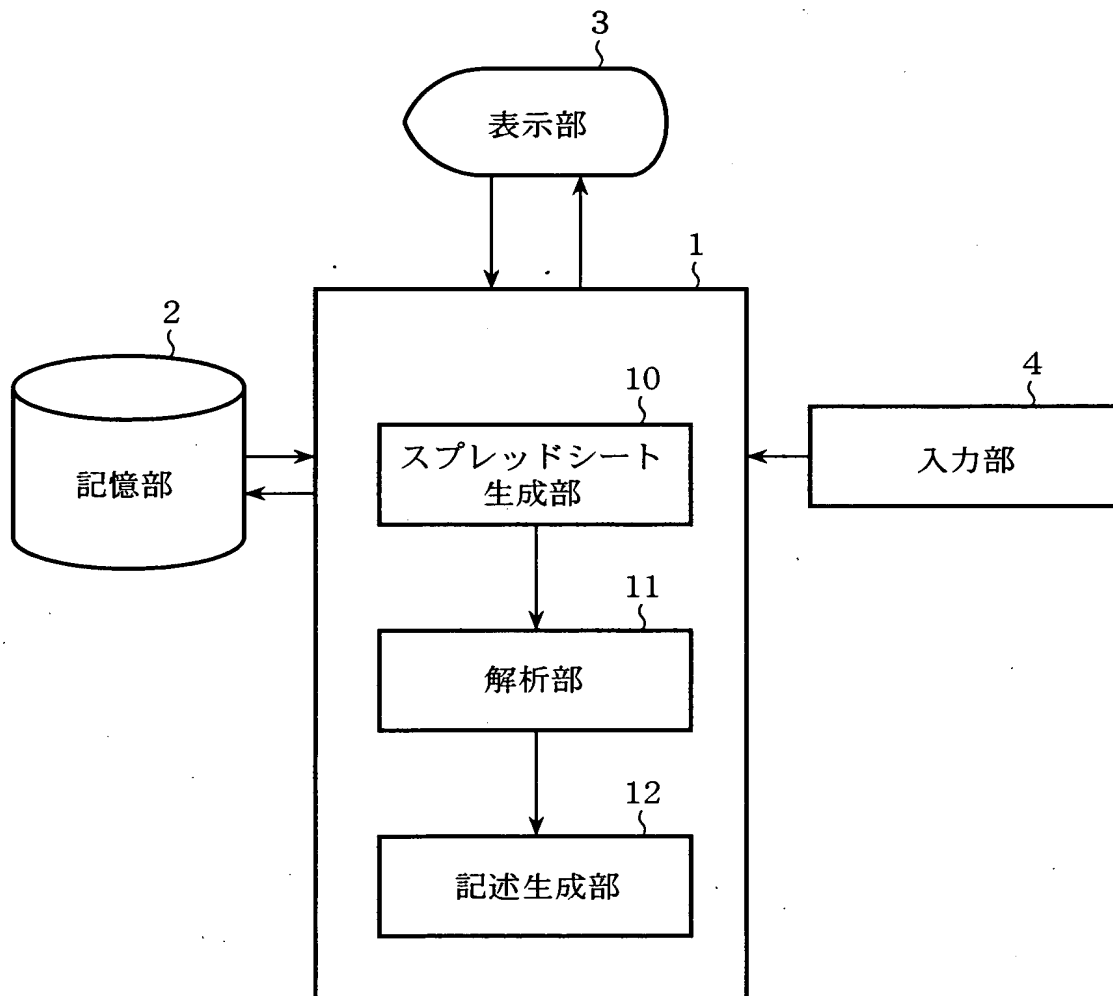
【図 2 0】 実施例で作成した、他の I P C 内の内部部品に関する接続条件とこれらの接続条件に応じた各内部部品の接続関係とを記述した下位階層の一覧表である。

【符号の説明】

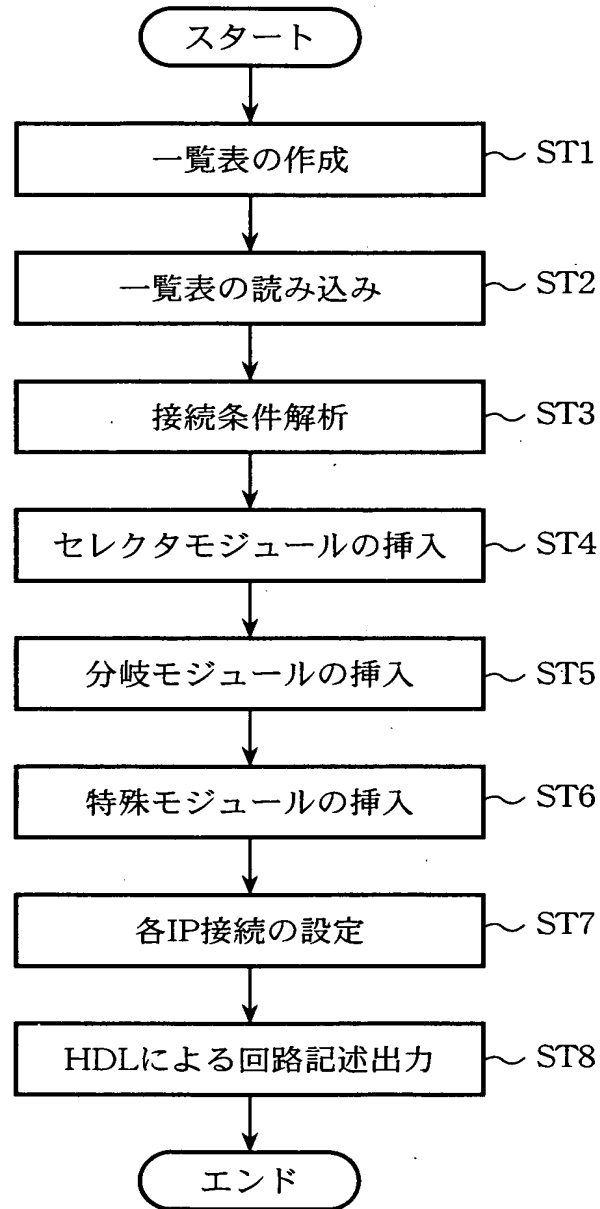
1 プロセッサ、2 記憶部、3 表示部、4 入力部、10 スプレッドシート生成部、11 解析部、12 記述生成部、20, 21 エンコーダモジュール（制御信号生成モジュール）、31, 32 回路部品モジュール、34, 43 ANDゲートモジュール、36 エンコーダモジュール（ゲート制御モジュール）、35, 42 分岐モジュール、44 ANDゲートモジュール（特殊モジュール）、A, B, C, 30, 40, 41 IP, BA, BB 信号線、BD 境界、BL1, BL2 ブロック、CP 制御パス、IPA1～IPA4, IPB1, IPB2, IPC1～IPC3 ピン、NP, NP1, NP2 通常パス、PA～PF ピン、SL1～SL6 セレクタモジュール、TP, TP1～TP5 テストパス。

【書類名】 図面

【図 1】



【図 2】

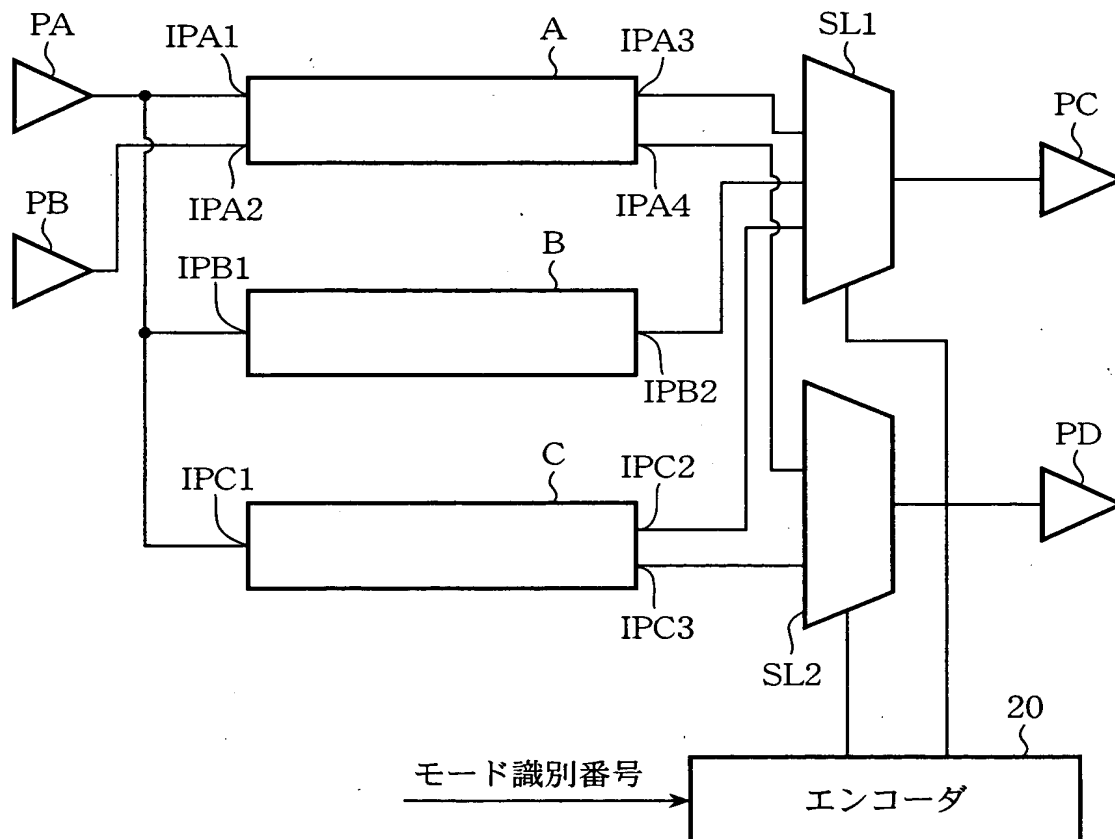




【図 3】

ピン名	接続条件 1	接続条件 2	接続条件 3
PA	IPA1	IPB1	IPC1
PB	IPA2	IPA2	IPA2
PC	IPA3	IPB2	IPC2
PD	IPA4	IPA4	IPC3

【図 4】



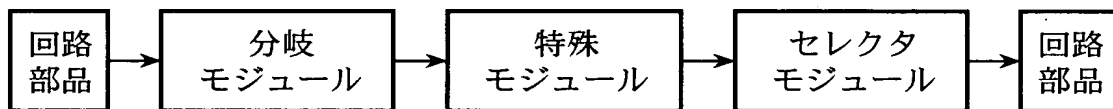
【図 5】

```

module TOP ( PA, PB, PC, PD );
  input PA, PB ;
  output PC, PD;
  if ( 接続条件1 == 1'b1 ) begin
    assign PA := IPA1 ;
  end

```

【図 6】



【図 7】

```

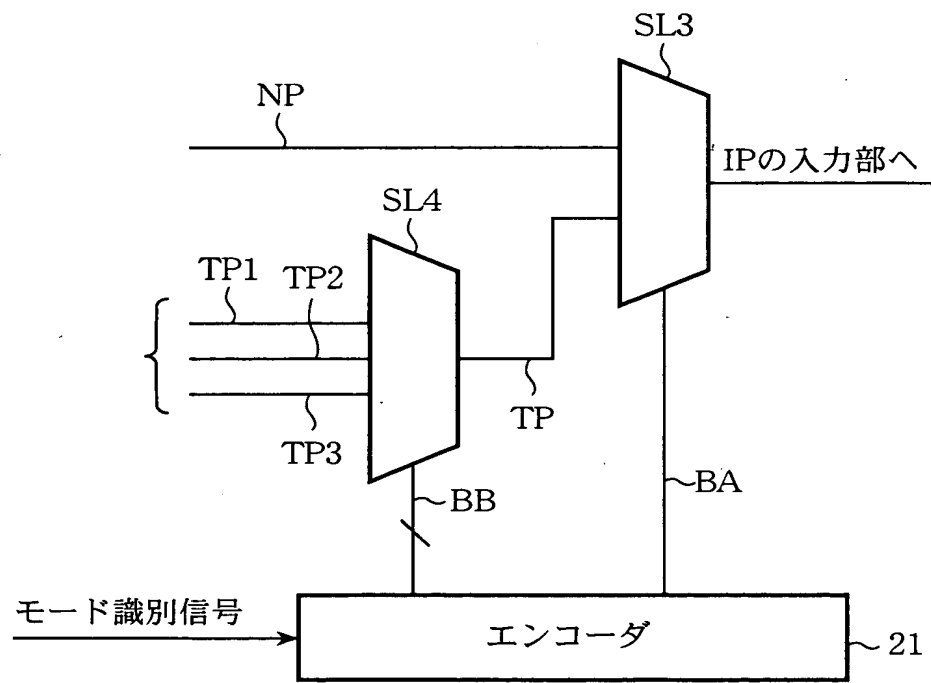
if ( 条件論理式1 )
  モード識別信号 1 = アクティブ
else
  モード識別信号 1 = 非アクティブ
end
if ( 条件論理式2 )
  モード識別信号 2 = アクティブ
else
  モード識別信号 2 = 非アクティブ
end

```

【図 8】

ピン名	通常使用 モード	テストモード 1	テストモード 2	テストモード 3

【図 9】



【図 1 0】

```

if ( BA== 1)
    IPの入力 = NP
else
    IPの入力 = セレクタモジュールSL4の出力
end
if ( BB == 1 )
    セレクタモジュールSL4の出力 = テストパスTP1
else if ( BB == 2 )
    セレクタモジュールSL4の出力 = テストパスTP2
else
    セレクタモジュールSL4の出力 = テストパスTP3
end

```

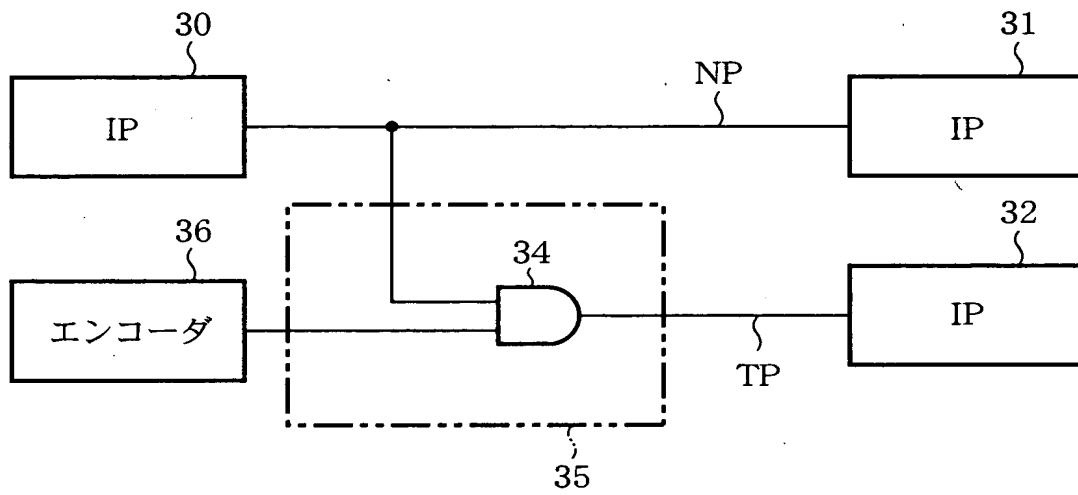
【図 1 1】

```

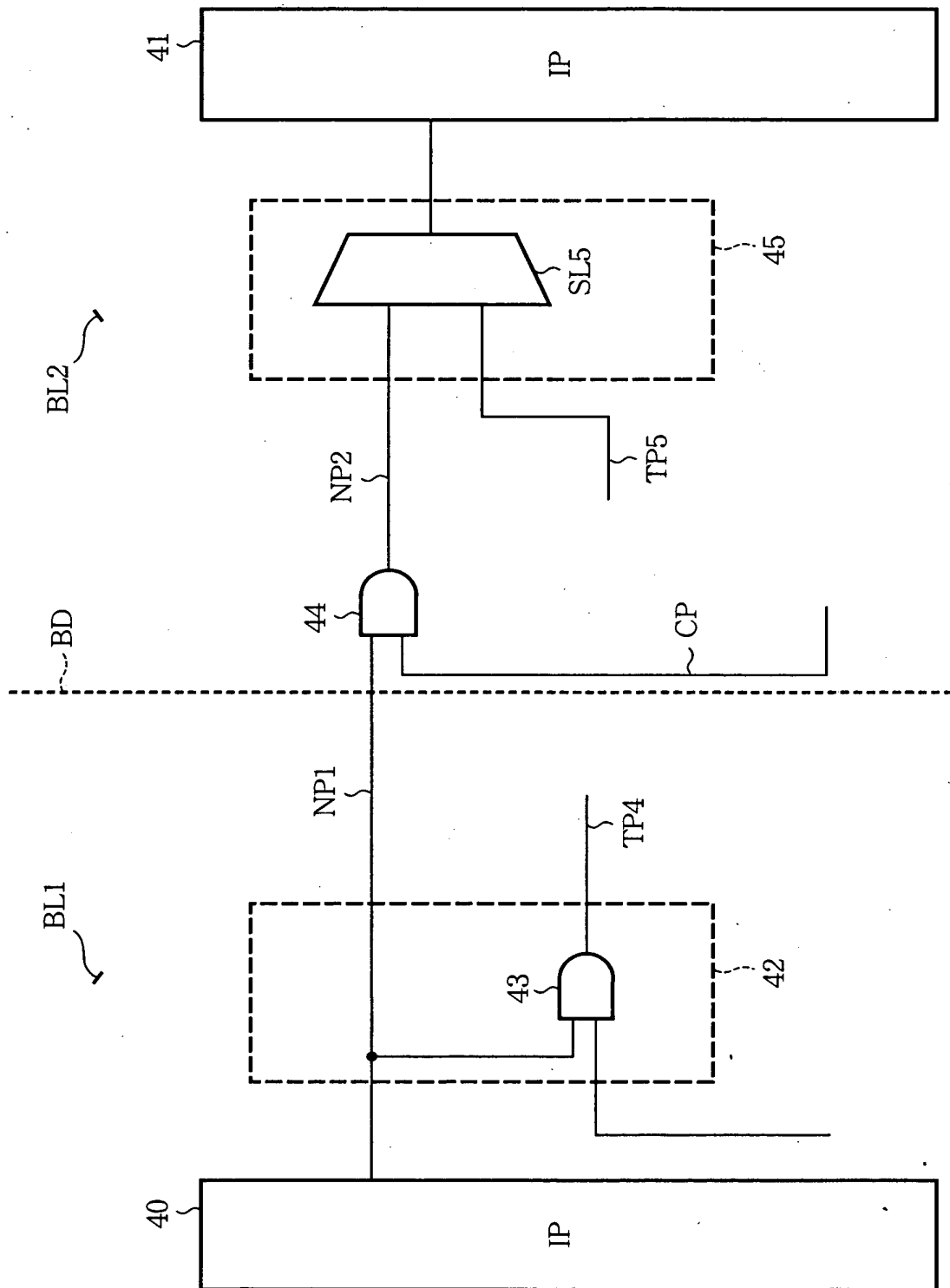
if ( モード識別信号 1 == 1 )
    BA = 通常パス選択制御
    BB = テストパスTP1選択制御
else if ( モード識別信号 2 == 1 )
    BA = テストパス選択制御
    BB = テストパスTP1選択制御
else if ( モード識別信号 3 == 1 )
    BA = テストパス選択制御
    BB = テストパスTP2選択制御
else
    BA = テストパス選択制御
    BB = テストパスTP3選択制御
end

```

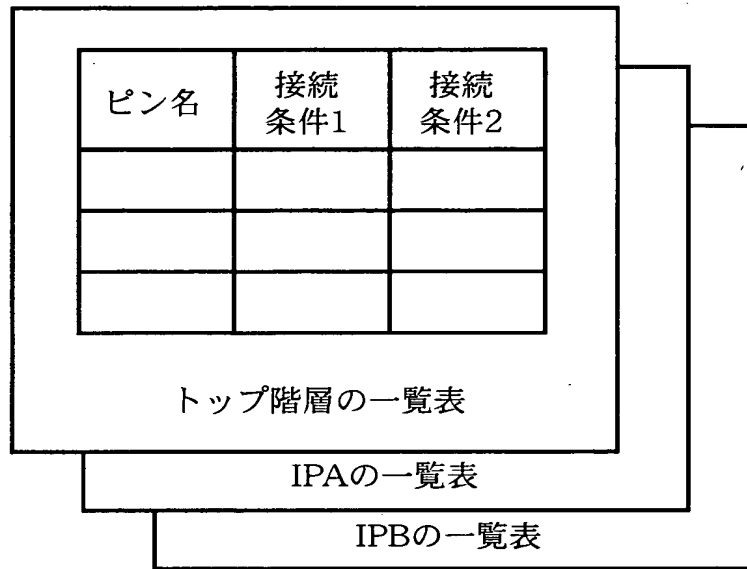
【図12】



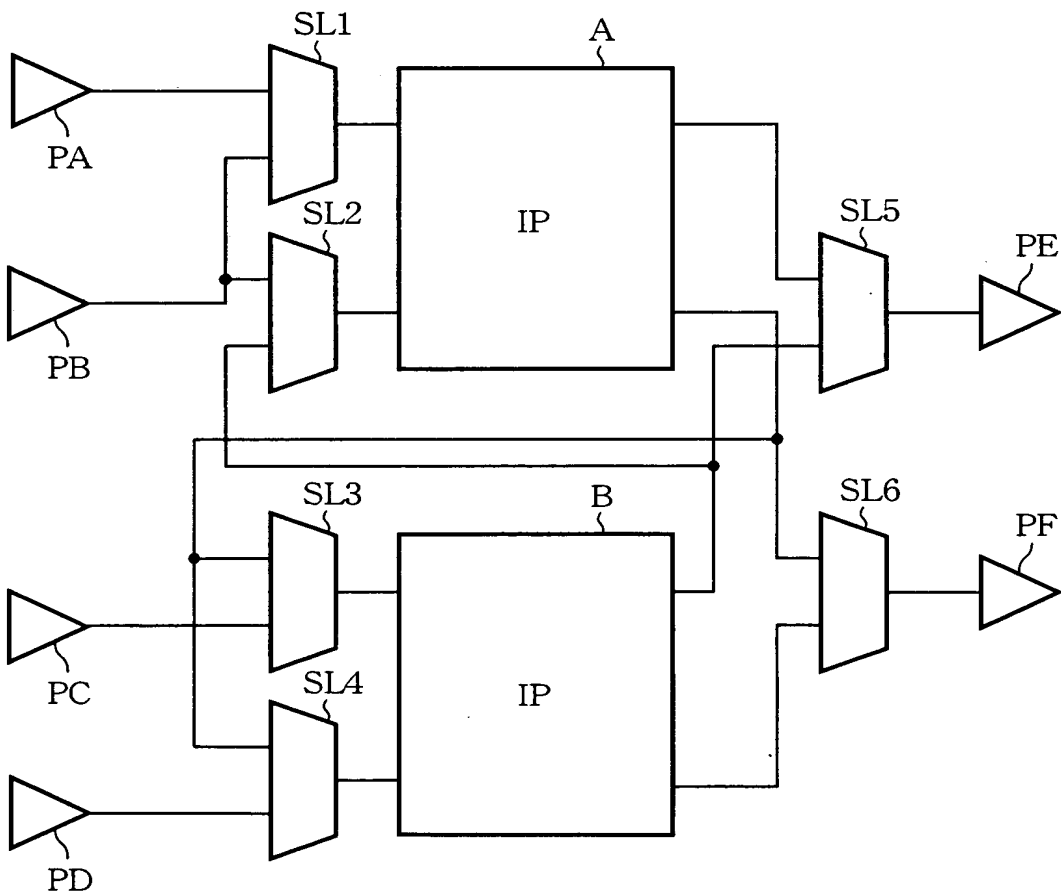
【図13】



【図14】



【図15】



【図 16】

電源グループ

A      ON

モード設定PIN		
PAD	T1	
PAD	T2	
PAD	C	

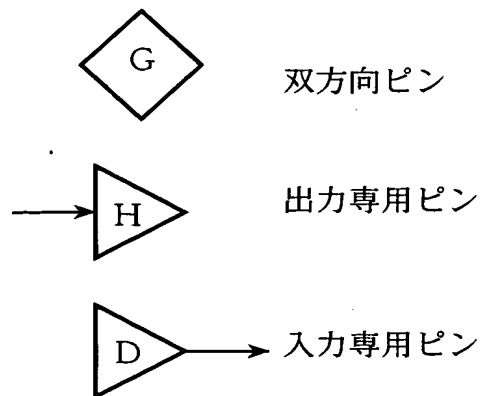
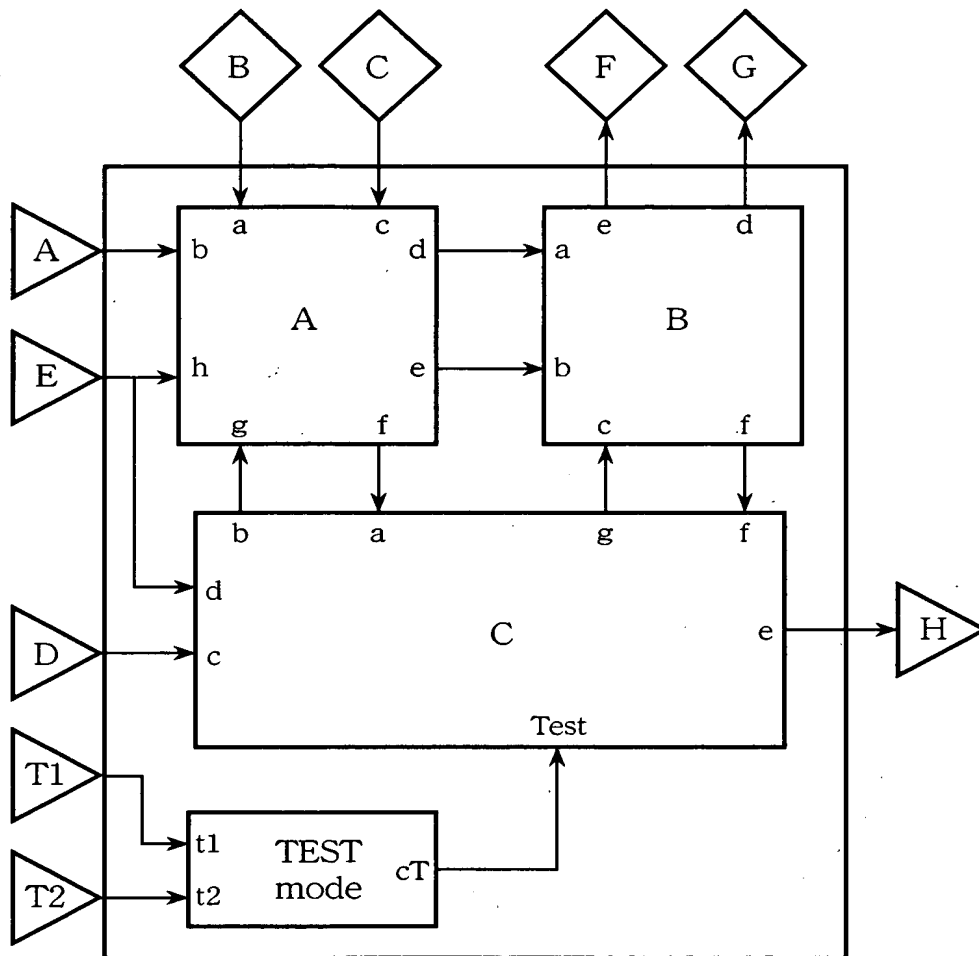
ブロック

PIn	IO	Default
A	Y	HUDS100LP
B	A	L4
↑	Y	U200
↑	C	L
↑	IE	L
C	A	H4
↑	Y	U200
↑	C	L
↑	IE	L
D	Y	L
E	Y	L
↑	↑	↑
F	A	L4
↑	Y	D200
↑	C	L
↑	IE	L
G	A	H4
↑	Y	D200
↑	C	L
↑	IE	L
H	A	L2
T1	Y	L
T2	Y	L

通常モード設定			A切り出しモード設定			B切り出しモード設定			C切り出しモード設定1			C切り出しモード設定2		
IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO
A	b	I	A	b	I	B	a	I	C	a	I	C	a	I
L	-	-	L	-	-	L	-	-	C	b	O	C	b	O
A	a	I	A	a	I	-	-	-	-	-	-	-	-	-
H	-	-	H	-	-	H	-	-	L	-	-	L	-	-
H	-	-	H	-	-	H	-	-	L	-	-	L	-	-
L	-	-	L	-	-	L	-	-	L	-	-	L	-	-
A	c	I	A	c	I	-	-	-	MODE	subT	I	MODE	subT	I
H	-	-	H	-	-	L	-	-	L	-	-	L	-	-
H	-	-	H	-	-	L	-	-	L	-	-	L	-	-
C	c	I	A	g	I	B	c	I	C	c	I	C	c	I
C	d	I	A	h	I	B	b	I	C	d	I	C	d	I
A	h	I	-	-	-	-	-	-	-	-	-	-	-	-
B	e	O	A	d	O	B	e	O	C	g	O	C	g	O
-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
A	FCL	O	L	-	-	L	-	-	L	-	-	L	-	-
A	FCL	O	L	-	-	L	-	-	L	-	-	L	-	-
B	d	O	A	e	O	B	d	O	L	-	-	L	-	-
-	-	-	-	-	-	-	-	-	C	f	I	C	f	I
L	-	-	L	-	-	L	-	-	H	-	-	H	-	-
L	-	-	L	-	-	L	-	-	H	-	-	H	-	-
C	e	O	A	f	O	B	f	O	C	e	O	C	e	O
MODE	t1	I	MODE	t1	I	MODE	t1	I	MODE	t1	I	MODE	t1	I
MODE	t2	I	MODE	t2	I	MODE	t2	I	MODE	t2	I	MODE	t2	I



【図 17】



【図 1 8】

電源グループ

B

ON-OFF

モード設定PIN												
PAD	T1											
PAD	T2											
PAD	C											
ブロック												
PIN	IO	Default										
a	I	L										
b	I	H										
c	I	L										
d	O	-										
e	O	-										
f	O	-										
g	I	H										
FCL	O	-										
h	I	L										

通常モード設定		A切り出しモード設定				B切り出しモード設定				C切り出しモード設定1				C切り出しモード設定2	
	0	0		1		1		1		1		1		1	
	0	1		0		0		1		1		1		1	
	-	-		-		-		0		0		1		1	
IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IO
PAD	B	I	PAD	B	I	L	-	-	L	-	-	L	-	-	-
PAD	A	I	PAD	A	I	L	-	-	L	-	-	L	-	-	-
PAD	C	I	PAD	C	I	L	-	-	L	-	-	L	-	-	-
B	a	O	PAD	F	O	-	-	-	-	-	-	-	-	-	-
B	b	O	PAD	G	O	-	-	-	-	-	-	-	-	-	-
C	a	O	PAD	H	O	-	-	-	-	-	-	-	-	-	-
C	b	I	PAD	D	I	L	-	-	L	-	-	L	-	-	-
PAD	F	O	-	-	-	-	-	-	-	-	-	-	-	-	-
PAD	E	I	PAD	E	I	L	-	-	L	-	-	L	-	-	-

【図 1 9】

電源グループ

B

ON-OFF

モード設定PIN			
PAD	T1		
PAD	T2		
PAD	C		
ブロック			
PIN	IO	Default	
e	O	-	
d	O	-	
a	I	L	
b	I	H	
c	I	L	
f	O	-	

通常モード設定		A切り出しモード設定			B切り出しモード設定			C切り出しモード設定1			C切り出しモード設定2			
IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO
PAD	F	O	-	-	-	PAD	F	O	-	-	-	-	-	-
PAD	G	O	-	-	-	PAD	G	O	-	-	-	-	-	-
A	d	I	L	-	-	PAD	A	I	L	-	-	L	-	-
A	e	I	L	-	-	PAD	E	I	L	-	-	L	-	-
C	g	I	-	-	*	PAD	D	I	-	-	*	-	-	*
C	f	O	-	-	-	PAD	H	O	-	-	-	-	-	-

【図 20】

電源グループ

A

ON

モード設定PIN		
PAD	T1	
PAD	T2	
PAD	C	

ブロック

PIN	IO	Default
d	I	L
c	I	H
e	O	-
a	I	L
b	O	-
f	I	L
g	O	-
test	I	L
PCA	O	-
PCB	O	-

通常モード設定			A切り出しモード設定			B切り出しモード設定			C切り出しモード設定1			C切り出しモード設定2		
0			0			1			1			1		
0			1			0			1			1		
-			-			-			0			1		
IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO	IP	PIN(IP)	IO
PAD	E	I	L	-	-	L	-	-	PAD	E	I	PAD	E	I
PAD	D	I	L	-	-	L	-	-	PAD	D	I	PAD	D	I
PAD	H	O	-	-	-	-	-	-	PAD	H	O	PAD	H	O
A	f	I	H	-	-	H	-	-	PAD	A	I	PAD	A	I
A	g	O	-	-	-	-	-	-	PAD	B	O	PAD	B	O
B	f	I	-	-	*	-	-	*	PAD	G	I	PAD	G	I
B	c	O	-	-	-	-	-	-	PAD	F	O	PAD	F	O
MODE	cT	I	MODE	cT	I	MODE	cT	I	MODE	cT	I	MODE	cT	I
MODE	AtoC	O	MODE	AtoC	O	MODE	AtoC	O	MODE	AtoC	O	MODE	AtoC	O
MODE	BtoC	O	MODE	BtoC	O	MODE	BtoC	O	MODE	BtoC	O	MODE	BtoC	O

【書類名】            要約書

【要約】

【課題】    複数の回路部品を接続した集積回路の自動設計を容易にし、ハードウェア記述言語による回路記述の作成にかかる手間と時間を短縮する。

【解決手段】    自動回路設計装置は、集積回路の複数の回路部品（例えば I P）に関する接続条件とこれらの接続条件に応じた各回路部品の接続関係とを記述した表形式のファイルを解析する解析部 1 1 と、解析部 1 1 の解析結果に基づいて、H D L による集積回路の記述を生成する記述生成部 1 2 とを備える。

【選択図】            図 1

出 願 人 履 歴 情 報

識別番号 [000006013]

1. 変更年月日 1990年 8月24日  
[変更理由] 新規登録  
住 所 東京都千代田区丸の内2丁目2番3号  
氏 名 三菱電機株式会社

出 願 人 履 歴 情 報

識別番号 [591036457]

1. 変更年月日 1991年 2月26日  
[変更理由] 新規登録  
住 所 東京都千代田区大手町2丁目6番2号  
氏 名 三菱電機エンジニアリング株式会社